Docket No. 1999-0078

NONPROVISIONAL PATENT APPLICATION

OLIFF & BERRIDGE, PLC
P.O. Box 19928
Alexandria, Virginia 22320
Telephone: (703) 836-6400
Facsimile: (703) 836-2787

Attorney Docket No.: 1999-0078

Date: April 13, 2000

BOX PATENT APPLICATION

Director of the U.S. Patent and Trademark Office
Washington, D.C. 20231

NONPROVISIONAL APPLICATION TRANSMITTAL RULE §1.53(b)

Sir:

Transmitted herewith for filing under 37 C.F.R. §1.53(b) is the nonprovisional patent application

For (Title): ITERATIVE DECODING

Casename:

By (Inventors): William TURIN

- ☒ Formal drawings (Figs. 1-9; nine sheets) are attached.
- ☒ A Declaration and Power of Attorney is filed herewith.
- ☒ An assignment of the invention to AT&T Corporation is filed herewith.
- ☒ An Information Disclosure Statement is filed herewith.
- ☐ A Preliminary Amendment is filed herewith.
- ☐ Please amend the specification by inserting before the first line the sentence --This nonprovisional application claims the benefit of U.S. Provisional Application No. _____, filed _____.--
- ☒ Priority of provisional application No. 60/174,601 filed January 5, 2000 is claimed under 35 U.S.C. §119.
- ☐ A certified copy of the above corresponding foreign application(s) is filed herewith.
- ☒ The filing fee is calculated below:

CLAIMS IN THE APPLICATION AFTER ENTRY
OF ANY PRELIMINARY AMENDMENT NOTED ABOVE

FOR:	NO. FILED	NO. EXTRA	RATE	FEE
BASIC FEE				$ 690
TOTAL CLAIMS	22 - 20	2*	x 18	$ 36
INDEP CLAIMS	2 - 3	0*	x 78	$
☐ MULTIPLE DEPENDENT CLAIMS PRESENTED			+260	$
* If the difference is less than zero, enter "0".			TOTAL	$ 726

- ☒ Check No. 107752 in the amount of $726 to cover the filing fee is attached. Except as otherwise noted herein, the Director is hereby authorized to charge any other fees that may be required to complete this filing, or to credit any overpayment, to Deposit Account No. 15-0461. Two duplicate copies of this sheet are attached.
- ☒ Please address all correspondence to S. H. Dworetsky, AT&T Corp., P.O. Box 4110, Middletown, New Jersey 07748. However, telephone calls should be made to Benjamin Lee at (973) 360-8117.

Respectfully submitted,

James A. Oliff, Reg. No. 27,075
Paul Tsou, Reg. No. 37,956
Attorney for Applicant(s)

JAO:PT/sdk
Date: April 13, 2000

OLIFF & BERRIDGE, PLC
P.O. Box 19928
Alexandria, Virginia 22320
Telephone: (703) 836-6400

DEPOSIT ACCOUNT USE AUTHORIZATION
Please grant any extension necessary for entry;
Charge any fee due to our
Deposit Account No. 15-0461

PT 13 8/92

ITERATIVE DECODING

This nonprovisional application claims the benefit of U.S. provisional application No. 60/174,601 entitled "Map Decoding In Channels With Memory" filed on January 5, 2000. The Applicant of the provisional application is William Turin (Attorney Docket No. 105038). The above provisional application is hereby incorporated by reference including all references cited therein.

BACKGROUND OF THE INVENTION

1. Field of Invention

This invention relates to iterative decoding of input sequences.

2. Description of Related Art

Maximum a posteriori (MAP) sequence decoding selects a most probable information sequence $X_1^T = (X_1, X_2, ..., X_T)$ that produced the received sequence $Y_1^T = (Y_1, Y_2, ..., Y_T)$. For transmitters and/or channels that are modeled using Hidden Markov Models (HMM), the process for obtaining the information sequence X_1^T that corresponds to a maximum probability is difficult due to a large number of possible hidden states as well as a large number of possible information sequences X_1^T. Thus, new technology is needed to improve MAP decoding for HMMs.

SUMMARY OF THE INVENTION

This invention provides an iterative process to maximum a posteriori (MAP) decoding. The iterative process uses an auxiliary function which is defined in terms of a complete data probability distribution. The MAP decoding is based on an expectation maximization (EM) algorithm which finds the maximum by iteratively maximizing the auxiliary function. For a special case of trellis coded modulation, the auxiliary function may be maximized by a combination of forward-backward and Viterbi algorithms. The iterative process converges monotonically and thus improves the performance of any decoding algorithm.

The MAP decoding decodes received inputs by minimizing a probability of error. A direct approach to achieve this minimization results in a complexity which grows exponentially with T, where T is the size of the input. The iterative process avoids this

complexity by converging on the MAP solution through repeated use of the auxiliary function.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is described in detail with reference to the following figures where like numerals reference like elements, and wherein:

Fig. 1 shows a diagram of a communication system;

Fig. 2 shows a flow chart of an exemplary iterative process;

Figs. 3-6 show state trajectories determined by the iterative process;

Fig. 7 shows an exemplary block diagram of the receiver shown in Fig. 1;

Fig. 8 shows a flowchart for an exemplary process of the iterative process for a TCM example; and

Fig. 9 shows step 1004 of the flowchart of Fig. 8 in greater detail.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Fig. 1 shows an exemplary block diagram of a communication system 100. The communication system 100 includes a transmitter 102, a channel 106 and a receiver 104. The transmitter 102 receives an input information sequence I_1^T (i.e., $I_1, I_2, ..., I_T$) of length T, for example. The input information sequence may represent any type of data including analog voice, analog video, digital image, etc. The transmitter 102 may represent a speech synthesizer, a signal modulator, etc.; the receiver 104 may represent a speech recognizer, a radio receiver, etc.; and the channel 106 may be any medium through which the information sequence X_1^T (i.e., $X_1, X_2, ..., X_T$) is conveyed to the receiver 104. The transmitter 102 may encode the information sequence I_1^T and transmit encoded information sequence X_1^T through the channel 106 and the receiver 104 receives information sequence Y_1^T (i.e., $Y_1, Y_2, ..., Y_T$). The problem in communications is, of course, to decode Y_1^T in such a way as to retrieve I_1^T.

Maximum a posteriori (MAP) sequence decoding is a technique that decodes the received sequence Y_1^T by minimizing a probability of error to obtain X_1^T (and if a model of the transmitter 102 is included, to obtain I_1^T). In MAP, the goal is to choose a most

probable X_1^T that produces the received Y_1^T. The MAP estimator may be expressed by equation 1 below.

$$\hat{X}_1^T = \underset{X_1^T}{\arg\max}\ \Pr(X_1^T, Y_1^T) \tag{1}$$

where $\Pr(\cdot)$ denotes a corresponding probability or probability density function and $\hat{X}_1^T$ is an estimate of X_1^T. Equation 1 sets $\hat{X}_1^T$ to the X_1^T that maximizes $\Pr(X_1^T, Y_1^T)$.

The $\Pr\left(X_1^T, Y_1^T\right)$ term may be obtained by modeling the channel 106 of the communication system 100 using techniques such as Hidden Markov Models (HMMs). An input-output HMM $\lambda = (S, \mathbf{X}, \mathbf{Y}, \pi, \{\mathbf{P}(X,Y)\})$ is defined by its internal states $S = \{1,2, \ldots n\}$, inputs X, outputs Y, initial state probability vector π, and the input-output probability density matrices (PDMs) $\mathbf{P}(X,Y)$, $X \in \mathbf{X}$, $Y \in \mathbf{Y}$. The elements of $\mathbf{P}(X,Y)$, $p_{ij}(X,Y) = \Pr(j,X,Y \mid i)$, are conditional probability density functions (PDFs) of input x and corresponding output y after transferring from the state i to state j. It is assumed that the state sequence $S_0^t = (S_0, S_1, \ldots, S_t)$, input sequence $X_1^t = (X_1, X_2, \ldots X_t)$, and output sequence $Y_1^t = (Y_1, Y_2, \ldots, Y_t)$ possess the following Markovian property

$$\Pr(S_t, X_t, Y_t | S_0^{t-1}, X_1^{t-1}, Y_1^{t-1}) = \Pr(S_t, X_t, Y_t | S_{t-1}).$$

Using HMM, the PDF of the input sequence X_1^T and output sequence Y_1^T may be expressed by equation 2 below:

$$p_T(X_1^T, Y_1^T) = \pi \prod_{i=1}^{T} \mathbf{P}(X_i, Y_i)\mathbf{1} \tag{2}$$

where $\mathbf{1}$ is a column vector of n ones, π is a vector of state initial probabilities, and n is a number of states in the HMM. Thus, the MAP estimator when using HMM may be expressed by equation 3 below:

$$\hat{X}_1^T = \underset{X_1^T}{\arg\max} \left[\pi \prod_{i=1}^{T} \mathbf{P}(X_i, Y_i)\mathbf{1} \right] \tag{3}$$

The maximization required by equation 3 is a difficult problem because all possible sequences of X_1^T must be considered. This requirement results in a complexity that grows exponentially with T. This invention provides an iterative process to obtain the maximum without the complexity of directly achieving the maximization by

evaluating equation 2 for all possible X_1^T, for example. In the iterative process, an auxiliary function is developed whose iterative maximization generates a sequence of estimates for X_1^T approaching the maximum point of equation 2.

The iterative process is derived based on the expectation maximization (EM) algorithm. Because the EM algorithm converges monotonically, the iterative process may improve the performance of any decoding algorithm by using its output as an initial sequence of the iterative decoding algorithm. In the following description, it is assumed that HMM parameters for the channel 106 and/or the transmitter 102 are available either by design or by techniques such as training.

The auxiliary function may be defined in terms of a complete data probability distribution shown in equation 4 below.

$$\Psi(z, X_1^T, Y_1^T) = \pi_{i_0} \prod_{i=1}^{T} p_{i_{t-1}i_t}(X_t, Y_t), \tag{4}$$

where $z = i_0^T$ is an HMM state sequence, π_{i_0} is an initial probability vector for state i_0, and $p_{ij}(X,Y)$ are the elements of the matrix $\mathbf{P}(X,Y)$. The MAP estimator of equation 1 can be obtained iteratively by equations 5-9 as shown below.

$$X_{1,p+1}^T = \arg\max_{X_1^T} Q(X_1^T, X_{1,p}^T), \quad p = 0,1,2,... \tag{5}$$

where p is a number of iterations and $Q(X_i^t, X_{i,p}^t)$ is the auxiliary function which may be expressed as

$$Q(X_1^T, X_{1,p}^T) = \sum_z \Psi(z, X_{1,p}^T, Y_1^T) \log\left(\Psi(z, X_1^T, Y_1^T)\right). \tag{6}$$

The auxiliary function may be expanded based on equation 4 as follows:

$$Q(X_1^T, X_{1,p}^T) = \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{t,ij}(X_{1,p}^T) \log(p_{ij}(X_t, Y_t)) + C \tag{7}$$

where C does not depend on X_1^T, n is a number of states in the HMM and

$$\gamma_{t,ij}(X_{1,p}^T) = \alpha_i(X_{1,p}^{t-1}, Y_1^{t-1}) p_{ij}(X_{t,p}, Y_t) \beta_j(X_{t+1,p}^T, Y_{t+1}^T) \tag{8}$$

where $\alpha_i(X_{1,p}^t, Y_1^t)$ and $\beta_j(X_{t+1,p}^T, Y_{t+1}^T)$ are the elements of the following forward and backward probability vectors

$$\alpha(X_1^t, Y_1^t) = \pi \prod_{i=1}^{T} P(X_i, Y_i), \text{ and } \beta(X_{t+1}^T, Y_{t+1}^T) = \prod_{i=t+1}^{T} P(X_i, Y_i)1. \quad (9)$$

Based on equations 5-9, the iterative process may proceed as follows. At p=0, an initial estimate of $X_{1,0}^T$ is generated. Then, $Q(X_1^T, X_{1,0}^T)$ is generated for all possible sequences of X_1^T. From equations 7 and 8, $Q(X_1^T, X_{1,0}^T)$ may be evaluated by generating $\gamma_{t,ij}(X_{1,0}^T)$ and $\log(p_{ij}(X_t, Y_t))$ for each t, i, and j. $\gamma_{t,ij}(X_{1,0}^T)$ may be generated by using the forward-backward algorithm as shown below:

$$\alpha(X_{1,p}^0, Y_1^0) = \pi, \quad \alpha(X_{1,p}^t, Y_1^t) = \alpha(X_{1,p}^{t-1}, Y_1^{t-1}) \mathbf{P}(X_{t,p}, Y_t) \quad ,t=1,2,\ldots T$$

$$\beta(X_{T+1,p}^T, Y_{T+1}^T) = \mathbf{1}, \quad \beta(X_{t,p}^T, Y_t^T) = \mathbf{P}(X_{t,p}, Y_t)\, \beta(X_{t+1,p}^T, Y_{t+1}^T) \quad ,t=T-1,T-2,\ldots,1$$

$\mathrm{Log}(p_{ij}(X_t, Y_t))$ is generated for all possible X_t for t=1, 2, ..., T and the (X_t)s that maximize $D(X_1^T, X_{1,0}^T)$ are selected as $X_{1,1}^T$. After $X_{1,1}^T$ is obtained, it is compared with $X_{1,0}^T$. If a measure $D(X_{1,1}^T, X_{1,0}^T)$ of difference between the sequences exceeds a compare threshold, then the above process is repeated until the difference measure $D(X_{1,p}^T, X_{1,p-1}^T)$ is within the threshold. The last $X_{1,p}^T$ for p iterations is the decoded output. The measure of difference may be an amount of mismatch information. For example, if X_1^T is a sequence of symbols, then the measure may be a number of different symbols between $X_{1,p}^T$ and $X_{1,p-1}^T$ (Hamming distance); if X_1^T is a sequence of real numbers, then the measure may be an Euclidean distance $D(X_{1,p}^T, X_{1,p-1}^T) = [\sum_{i=1}^{T} (X_{i,p} - X_{i,p-1})^2]^{1/2}$.

Fig. 2 shows a flowchart of the above-described process. In step 1000, the receiver 104 receives the input information sequence Y_1^T and goes to step 1002. In step 1002, the receiver 104 selects an initial estimate for the decode output information sequence $X_{1,0}^T$ and goes to step 1004. In step 1004, the receiver 104 generates $\gamma_{t,ij}(X_{1,p}^T)$ where p = 0 for the first iteration and goes to step 1006. In step 1006, the receiver 104 generates all the $\log(p_{ij}(X_t, Y_t))$ values and goes to step 1008.

In step 1008, the receiver 104 selects a sequence $X_{1,p+1}^T$ that maximizes $Q(X_{1,p+1}^T, X_{1,p}^T)$ and goes to step 1010. In step 1010, the receiver 104 compares $X_{1,p}^T$ with

$X^T_{1,p+1}$. If the compare result is within the compare threshold, then the receiver 104 goes to step 1012; otherwise, the receiver 104 returns to step 1004 and continues the process with the new sequence $X^T_{1,p+1}$. In step 1012, the receiver 104 outputs $X^T_{1,p+1}$ and goes to step 1014 and ends the process.

The efficiency of the above described iterative technique may be improved if the transmitted sequence is generated by modulators such as a trellis coded modulator (TCM). A TCM may be described as a finite state machine that may be defined by equations 10 and 11 shown below.

$$S_{t+1} = f_t(S_t, I_t) \tag{10}$$

$$X_t = g_t(S_t, I_t) \tag{11}$$

Equation 10 specifies the TCM state transitions while equation 11 specifies the transmitted information sequence based on the state and the input information sequence. For example, after receiving input I_t in state S_t, the finite state machine transfers to state S_{t+1} based on S_t and I_t as shown in equation 10. The actual output by the transmitter 102 is X_t according to equation 11. Equation 10 may represent a convolutional encoder and equation 11 may represent a modulator. For the above example, the transmitter output information sequence X_1^T may not be independent even if the input information sequence I_1^T is independent.

In equation 15, the $\log(p_{ij}(Y_t,X_t))$ term may be analyzed based on the TCM state transitions because the information actually transmitted X_t is related to the source information I_t by $X_t = g_t(S_t, I_t)$. This relationship between X_t and I_t forces many elements $p_{ij}(Y_t,X_t)$ of $P(Y_t,X_t)$, to zero since the finite state machine (equations 10 and 11) removes many possibilities that otherwise must be considered. Thus, unlike the general case discussed in relation to equations 5-9, evaluation of $p_{ij}(Y_t,X_t)$ may be divided into a portion that is channel related and another portion that is TCM related. The following discussion describes the iterative technique in detail for the TCM example.

For a TCM system with an independent and identically distributed information sequence, an input-output HMM may be described by equations 12 and 13 below.

$$P(X_t, Y_t) = \left[p_{S_t S_{t+1}} \mathbf{P}_c(X_t \mid Y_t)\right], \tag{12}$$

where

$$p_{S_tS_{t+1}} = \begin{cases} \Pr(I_t) \text{ if } S_{t+1} = f_t(S_t, I_t) \\ 0 \quad \text{otherwise} \end{cases} \tag{13}$$

$\mathbf{P}_c(Y_t \mid X_t)$ is the conditional PDM of receiving Y_t given that X_t has been transmitted for the HMM of a medium (channel) through which the information sequence is transmitted; $p_{S_tS_{t+1}}$ is the probability of the TCM transition from state S_t to state S_{t+1}, and $\Pr(I_t)$ is the probability of an input I_t. Thus, equation 2 may be written as

$$p_T(I_1^T, Y_1^T) = \pi_c \prod_{i=1}^{T} p_{s_ts_{t+1}} \mathbf{P}_c(Y_t \mid X_t)\mathbf{1},$$

where π_c is a vector of the initial probabilities of the channel states, $X_t = g_t(S_t, I_t)$, and the product is taken along the state trajectory $S_{t+1} = f_t(S_t, I_t)$ for $t = 1, 2, \ldots, T$.

If all elements of the input information sequence are equally probable, then the MAP estimate may be expressed by equation 14 below.

$$\hat{I}_1^T = \arg\max_{I_1^T} \pi_c \prod_{t=1}^{T} \mathbf{P}_c(Y_t \mid X_t)\mathbf{1}, \tag{14}$$

The auxiliary function may be expressed by equations 15-17 below corresponding to equations 7-9 above.

$$Q(I_1^T, I_{1,p}^T) = \sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{j=1}^{n} \gamma_{t,ij}(I_{1,p}^T)\log(p_{ij}(Y_t \mid X_t)) + C \tag{15}$$

where $X_t = g_t(S_t, I_t)$ and

$$\gamma_{t,ij}(I_{1,p}^T) = \alpha_i(Y_1^{t-1} \mid I_{1,p}^t) p_{c,ij}(Y_t \mid X_{t,p}) \beta_j(Y_{t+1,p}^T \mid I_{t+1,p}^T) \tag{16}$$

$\alpha_i(Y_1^{T-1} \mid I_{1,p}^{t-1})$ and $\beta_j(Y_{t+1,p}^T \mid I_{t+1,p}^T)$ are the elements of the forward and backward probability vectors

$$\alpha(Y_1^t \mid I_1^t) = \pi_c \prod_{i=1}^{t} \mathbf{P}_c(Y_i \mid X_i) \text{ and } \beta(Y_{t+1}^T \mid I_{t+1}^T) = \prod_{i=t+1}^{T} \mathbf{P}_c(Y_i \mid X_i)\mathbf{1} \tag{17}$$

From equation 15, the Viterbi algorithm may be applied with the branch metric

$$m(I_t) = \sum_{i=1}^{n}\sum_{j=1}^{n} \gamma_{t,ij}(I_{1,p}^T)\log p_{c,ij}(Y_t \mid X_t), \ t = 1, 2, \ldots, T \tag{18}$$

to find a maximum of $Q(I_1^T, I_{1,p}^T)$ which can be interpreted as a longest path leading from the initial zero state to one of the states S_T where only the *encoder* trellis is considered. The Viterbi algorithm may be combined with the backward portion of the forward-backward algorithm as follows.

1. Select an initial source information sequence $I_{1,0}^T = I_{1,0}, I_{2,0}, \ldots, I_{T,0}$
2. *Forward part:*
   a. set $\alpha(Y_1^0 \mid I_1^0) = \pi$, where π is an initial state probability estimate;

and

   b. for $t = 1, 2, \ldots, T$, compute $X_{t,p} = g_t(S_t, I_{t,p})$,
   $\alpha(Y_1^t \mid I_{1,p}^t) = \alpha(Y_1^{t-1} \mid I_{1,p}^{t-1}) \mathbf{P}_c(Y_t \mid X_{1,p})$,
   where $I_{1,p}^t$ is a prior estimate of I_1^t.
3. *Backward part:*
   a. set $\beta\left(Y_{T+1}^T \mid I_{T+1,p}^T\right) = \mathbf{1}$ and last state transition lengths $L(S_T)$ to 0

for all the states;

   for $t = T, T\text{-}1, \ldots, 1$ compute:

   b. $X_t = g_t(S_t, I_t)$,
   c. $\gamma_{t,ij}\left(I_{1,p}^T\right) = \alpha_i\left(Y_1^{t-1} \mid I_{1,p}^{t-1}\right) p_{c,j}\left(Y_t \mid X_{t,p}\right) \beta_j\left(Y_{t+1}^T \mid I_{t+1,p}^T\right)$,
   d. $L(S_t) = \max_{I_t}\{L[f_t(S_t, I_t)] + m(I_t)\}$. This step selects the paths with the largest lengths (the survivors).
   e. $\hat{I}_t(S_t) = \arg\max_{I_t}\{L[f_t(S_t, I_t)] + m(I_t)\}$. This step estimates I_t corresponding to the state S_t by selecting the I_t of the survivor in step d.
   f. $\beta(Y_t^T \mid I_{t,p}^T) = \mathbf{P}_c(Y_t \mid X_{t,p})\, \beta(Y_{t+1}^T \mid X_{t+1,p}^T)$.
   g. End (of "for" loop).
4. Reestimate the information sequence:

$I_{t,p+1} = \hat{I}_t(\hat{S}_t)$, $\hat{S}_{t+1} = f_t(\hat{S}_t, I_{t,p+1})$, $t = 1,2,\ldots,T$ where $\hat{S}_1 = 0$; and

5. If $I_{t,p+1} \neq I_{t,p}$, go to step 2; otherwise decode the information sequence as $I^T_{t,p+1}$.

Figs. 3-6 show an example of the iterative process discussed above where there are four states in the TCM and T = 5. The dots represent possible states and the arrows represent a state trajectory that corresponds to a particular information sequence. The iterative process may proceed as follows. First, an initial input information sequence $I^5_{1,0}$ is obtained. $I^5_{1,0}$ may be the output of an existing decoder or may simply be a guess.

The Viterbi algorithm together with the backward algorithm may be used to obtain a next estimate of the input information sequence $I^5_{1,1}$. This process begins with the state transitions between t = 4 and t = 5 by selecting state transitions leading to each of the states s0-s3 at t = 4 from states at t = 5 that have the largest value of the branch metric $L(S_4) = m(I_5)$ of equation 18 above. Then, the process moves to select state transitions between the states at t = 3 and t = 4 that have the largest cumulative distance $L(S_3) = L(S_4) + m(I_4)$. This process continues until t = 0 and the sequence of input information I^5_1 corresponding to the path connecting the states from t = 0 to t = 5 that has the longest path $L(S_0) = \sum_{t=1}^{5} m(I_t)$ is selected as the next input information sequence $I^5_{1,1}$.

For the example in Fig. 3, state transitions from the states at t = 4 to all the states at t = 5 are considered. Assuming that the (I_t)s are binary, then only two transitions can emanate from each of the states at t = 4: one transition for $I_5 = 0$ and one transition for $I_5 = 1$. Thus, Fig. 3 shows two arrows terminating on each state at t = 4 (arrows are "backwards" because the backward algorithm is used). State transitions 301 and 302 terminate at state s0; state transitions 303 and 304 terminate at state s1; state transitions 305 and 306 terminate at state s2; and state transitions 307 and 308 terminate at state s3.

The branch metric $m(I_t)$ of equation 18 represents a "distance" between the states and is used to select the state transition that corresponds to the longest path for each of the states s0-s3 at t = 4:

$$m(I_5) = \sum_i \sum_j \gamma_{5,ij}(X_{1,0}^5) \log p_{ij}(X_5, Y_5)$$

$$= \sum_i \sum_j \alpha_i(X_{1,0}^4, Y_1^4) p_{ij}(X_{5,0}, Y_5) \beta_j(X_{6,0}^5, Y_6^5) \log p_{ij}(Y_5 | X_5), \qquad (19)$$

where $\beta_j(X_{6,0}^5, Y_6^5) = 1$, and $X_5 = g_5(S_5, I_5)$ by definition. There is an I_5 that corresponds to each of the state transitions 301-308. For this example, $L(S_4) = m(I_5)$ corresponding to odd numbered state transitions 301-307 are greater than that for even numbered state transitions 302-308. Thus, odd numbered state transitions are "survivors." Each of them may be part of the state trajectory that has the longest path from $t = 0$ to $t = 5$. This transition (the survivor) is depicted by the solid arrow while the transitions with smaller lengths are depicted by dashed lines.

The state sequence determination process continues by extending the survivors to $t = 3$ as shown in Fig. 4 forming state transitions 309-316. The distance between state transitions for each of the states are compared based on $L(S_4) + m(I_4)$, where $m(I_4)$ is shown in equation 20 below.

$$m(I_4) = \sum_i \sum_j \gamma_{5,ij}(X_{1,0}^5) \log p_{ij}(X_4, Y_4)$$

$$= \sum_i \sum_j \alpha_i(X_{1,0}^3, Y_1^3) p_{ij}(X_{4,0}, Y_4) \beta_j(X_{5,0}, Y_5) \log p_{ij}(Y_4 | X_4). \qquad (20)$$

For this example, the distances corresponding to the odd numbered state transitions 309-315 are longer than distances corresponding to even numbered state transitions 310-316. Thus, the paths corresponding to the odd numbered state transitions are the survivors. As shown in Fig. 4, the state transition 301 is not connected to any of the states at $t = 3$ and thus is eliminated even though it was a survivor. The other surviving state transitions may be connected into partial state trajectories. For example, partial state trajectories are formed by odd numbered state transitions 307-309, 303-311, 303-313 and 305-315.

The above process continues until $t = 0$ is reached as shown in Fig. 5 where two surviving state trajectories 320-322 are formed by the surviving state trajectories. All the state trajectories terminate at state zero for this example because, usually, encoders start at state zero. As shown in Fig. 6, the state trajectory that corresponds to the longest cumulative distance is selected and the input information sequence I_1^5 (via $S_{t+1} = f_t(S_t, I_t)$ that corresponds to the selected trajectory is selected as the next estimated input

information sequence $\hat{I}_{1,1}^{5}$. For this example, the state trajectory 320 is selected and the input information sequence I_1^5 corresponding to the state trajectory 320 is selected as $\hat{I}_{1,1}^{5}$.

Fig. 7 shows an exemplary block diagram of the receiver 104. The receiver 104 may include a controller 202, a memory 204, a forward processor 206, a backward processor 208, a maximal length processor 210 and an input/output device 212. The above components may be coupled together via a signal bus 214. While the receiver 104 is illustrated using a bus architecture, any architecture may be suitable as is well known to one of ordinary skill in the art.

All the functions of the forward, backward and maximal length processors 206, 208 and 210 may also be performed by the controller 202 which may be either a general purpose or special purpose computer (e.g., DSP). Fig. 7 shows separate processors for illustration only. The forward, backward maximal length processors 206, 208 and 210 may be combined and may be implemented by using ASICs, PLAs, PLDs, etc. as is well known in the art.

The forward processor 206 generates the forward probability vectors $\alpha_i\left(X_{1,p}^{t-1}, Y_1^{t-1}\right)$ herein referred to as α_i. For every iteration, when a new $X_{1,p}^{T}$ (or $I_{1,p}^{T}$) is generated, the forward processor 206 may generate a complete set of α_i.

The backward processor 208 together with the maximal length processor 210 generate a new state sequence by searching for maximal length state transitions based on the branch metric $m(I_t)$. Starting with the final state transition between states corresponding to $t = T - 1$ and $t = T$, the backward processor generates $\beta_j(X_{t+1,p}^{T}, Y_{t+1}^{T})$ (hereinafter referred as β_j) as shown in equation 8 for each state transition.

The maximal length processor 210 generates $m(I_t)$ based on the results of the forward processor 206, the backward processor 208 and $p_{ij}(X_t,Y_t)$. After generating all the $m(I_t)$s corresponding to each of the possible state transitions, the maximal length processor 210 compares all the $L(S_t) + m(I_t)$s and selects the state transition that corresponds to the largest $L(S_t) + m(I_t)$, and the I_t (via $S_{t+1} = f_t(S_t, I_t)$) that corresponds to the selected state transition is selected as the estimated input information for that t. The

above process is performed for each t = 1, 2, ..., T to generate a new estimate $I_{1,p}^T$ for each of the iteration p.

Initially, the controller 202 places an estimate of the PDM **P**(X,Y) and π in the memory 204 that corresponds to the HMM for the channel 106 and/or the transmitter 102. The PDM **P**(X,Y) may be obtained via well known training processes, for example.

When ready, the controller 202 receives the received input information sequence Y_1^T and places them in the memory 204 and selects an initial estimate of $I_{1,0}^T$ (or $X_{1,0}^T$). The controller 202 coordinates the above-described iterative process until a new estimate $I_{1,1}^T$ (or $X_{1,1}^T$) is obtained. Then, the controller 202 compares $I_{1,0}^T$ with $I_{1,1}^T$ to determine if the compare result is below the compare threshold value (e.g., matching a predetermined number of elements or symbols of the information sequence). The compare threshold may be set to 0, in which case $I_{1,0}^T$ must be identical with $I_{1,1}^T$. If an acceptable compare result is reached, $I_{1,1}^T$ is output as the decoded output. Otherwise, the controller 202 iterates the above-described process again and compares the estimated $I_{1,p}^T$ with $I_{1,p-1}^T$ until an acceptable result is reached and $I_{1,p}^T$ is output as the decoded output.

Fig. 8 shows a flowchart of the above-described process. In step 1000, the controller 202 receives Y_1^T via the input/output device 212 and places Y_1^T in the memory 204 and goes to step 1002. In step 1002, the controller 202 selects an initial estimate for $I_{1,0}^T$ and goes to step 1004. In step 1004, the controller 202 determines a new state sequence and a next estimated $I_{1,1}^T$ ($I_{1,p}^T$, where p = 1) (via the forward, backward and maximal length processors 206, 208 and 210) and goes to step 1006. In step 1006, the controller 202 compares $I_{1,0}^T$ with $I_{1,1}^T$. If the compare result is within the predetermined threshold, then the controller 202 goes to step 1008; otherwise, the controller 202 returns to step 1004. In step 1008, the controller 202 outputs $I_{1,p}^T$ where p is the index of the last iteration and goes to step 1010 and ends the process.

Fig. 9 shows a flowchart that expands step 1004 in greater detail. In step 2000, the controller 202 instructs the forward processor 206 to generate α_i as shown in

equation 8, and goes to step 2002. In step 2002, the controller 202 sets the parameter $t = T$ and goes to step 2004. In step 2004, the controller 202 instructs the backward processor 208 to generate β_j and the maximal length processor 210 to determine next set of survivors based on equation 18 and time t+1 survivors and goes to step 2006.

In step 2006, the controller 202 decrements t and goes to step 2008. In step 2008, the controller 202 determines whether t is equal to 0. If t is equal to 0, the controller 202 goes to step 2010; otherwise, the controller 202 returns to step 2004. In step 2010, the controller 202 outputs the new estimated I_1^T and goes to step 2012 and returns to step 1006 of Fig. 5.

A specific example of the iterative process for convolutional encoders is enclosed in the appendix.

While this invention has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, modifications, and variations will be apparent to those skilled in the art. Accordingly, preferred embodiments of the invention as set forth herein are intended to be illustrative, not limiting. Various changes may be made without departing from the spirit and scope of the invention.

For example, a channel may be modeled as $\mathbf{P}_c(Y \mid X) = \mathbf{P}_c\mathbf{B}_c(Y \mid X)$ where $\mathbf{P}_c$ is a channel state transition probability matrix and $\mathbf{B}_c(Y \mid X)$ is a diagonal matrix of state output probabilities. For example, based on the Gilbert-Elliott model

$$\mathbf{B}_c(X \mid X) = \begin{bmatrix} 1-b_1 & 0 \\ 0 & 1-b_2 \end{bmatrix} \text{ and } \mathbf{B}_c(\overline{X} \mid X) = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix},$$

where $\overline{X}$ is the complement of X. For this case, $m(I_t)$ may be simplified as

$$m(I_t) = \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) b_j(Y_t \mid X_t),\ t = 1, 2, \ldots, T, \text{ and}$$

$\gamma_{t,i}(I_{1,p}^T) = \alpha_i(Y_1^t \mid I_{1,p}^t)\beta_i(Y_{t+1}^T \mid I_{t+1,p}^T)$, where $b_j(Y_t \mid X_t)$ are the elements of $\mathbf{B}_c$.

WHAT IS CLAIMED IS:

1. A method for maximum a posteriori (MAP) decoding of an input information sequence based on a first information sequence received through a channel, comprising:

iteratively generating a sequence of one or more decode results starting with an initial decode result; and

outputting one of adjacent decode results as a decode of the input information sequence if the adjacent decode results are within a compare threshold.

2. The method of claim 1, wherein the iteratively generating comprises:

a. generating the initial decode result as a first decode result;

b. generating a second decode result based on the first decode result and a model of the channel;

c. comparing the first and second decode results;

d. replacing the first decode result with the second decode result; and

e. repeating b-d if the first and second decode results are not within the compare threshold.

3. The method of claim 2, wherein the generating a second decode result comprises searching for a second information sequence that maximizes a value of an auxiliary function.

4. The method of claim 3, wherein the auxiliary function is based on the expectation maximization (EM) algorithm.

5. The method of claim 4, wherein the model of the channel is a Hidden Markov Model (HMM) having an initial state probability vector π and probability density matrix (PDM) of $\mathbf{P}(X,Y)$, where $X \in \mathbf{X}$, $Y \in \mathbf{Y}$ and elements of $\mathbf{P}(X,Y)$, $p_{ij}(X,Y) = \Pr(j,X,Y \mid i)$, are conditional probability density functions of an information element X of the second information sequence that corresponds to a received element Y of the first information sequence after the HMM transfers from a state i to a state j, the auxiliary function being expressed as:

$$Q(X_1^T, X_{1,p}^T) = \sum_z \Psi(z, X_{1,p}^T, Y_1^T)\log\left(\Psi(z, X_1^T, Y_1^T)\right),$$ where p is a number of iterations, $\Psi(z,\ X_1^T,\ Y_1^T) = \pi_{i_0}\prod_{t=1}^{T} p_{i_{t-1}i_t}(X_t, Y_t)$, T is a number of information elements in a particular information sequence, z is a HMM state sequence i_0^T, π_{i_0} is the probability of an initial state i_0, X_1^T is the second information sequence, $X_{1,p}^T$ is a second information sequence estimate corresponding to a pth iteration, and Y_1^T is the first information sequence.

6. The method of claim 5, wherein the auxiliary function is expanded to be:

$$Q(X_1^T, X_{1,p}^T) = \sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{j=1}^{n} \gamma_{t,ij}(X_{1,p}^T)\log(p_{ij}(X_t, Y_t)) + C$$

where C does not depend on X_1^T and

$$\gamma_{t,ij}(X_{1,p}^T) = \alpha_i(X_{1,p}^{t-1}, Y_1^{t-1})p_{ij}(X_{t,p}, Y_t)\,\beta_j(X_{t+1,p}^T, Y_{t+1}^T)$$

where $\alpha_i(X_{1,p}^t, Y_1^T)$ and $\beta_j(X_{t+1,p}^T, Y_{t+1}^T)$ are the elements of forward and backward probability vectors defined as

$$\alpha(X_1^t,\ Y_1^t) = \pi\prod_{i=1}^{t} P(X_i, Y_i),\ \text{and}\ \beta(X_1^T, Y_1^T) = \prod_{j=t}^{T} P(X_j, Y_j)\mathbf{1},$$ π is an initial probability vector, **1** is the column vector of ones.

7. The method of claim 6, wherein a source of an encoded sequence is a trellis code modulator (TCM), the TCM receiving a source information sequence I_1^T and outputting X_1^T as an encoded information sequence that is transmitted, the TCM defining $X_t = g_t(S_t, I_t)$ where X_t and I_t are the elements of X_1^T and I_1^T for each time t, respectively, S_t is a state of the TCM at t, and $g_t(.)$ is a function relating X_t, to I_t and S_t, the method comprising:

generating, for iteration p+1, a source information sequence estimate $I_{1,p+1}^T$ that corresponds to a sequence of TCM state transitions that has a longest cumulative distance $L(S_{t-1})$ at t = 1 or $L(S_0)$, wherein a distance for each of the TCM state transitions is defined by $L(S_t) = L(S_{t+1}) + m(\hat{I}_{t+1}(S_{t+1}))$for the TCM state transitions at each t for

$t = 1, ..., T$ and the cumulative distance is the sum of $m(\hat{I}_t(S_t))$ for all t, $m(\hat{I}_t(S_t))$ being defined as

$$m\left(\hat{I}_t(S_t)\right) = \sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \gamma_{t,ij}\left(I_{1,p}^T\right) \log p_{c,ij}\left(Y_t \mid X_t(S_t)\right),$$ for each $t = 1, 2, ..., T$, where $X_t(S_t) = g_t(S_t, \hat{I}_t(S_t))$, n_c is a number of states in an HMM of the channel and $p_{c,ij}(Y_t \mid X_t(S_t))$ are channel conditional probability density functions of Y_t when $X_t(S_t)$ is transmitted by the TCM, $I_{1,p+1}^T$ being set to a sequence of $\hat{I}_t$ for all t.

8. The method of claim 7, wherein for each $t = 1, 2, ..., T$, the method comprises:

generating $m(\hat{I}_t(S_t))$ for each possible state transition of the TCM;

selecting state trajectories that correspond to largest $L(S_t) = L(S_{t+1}) + m(\hat{I}_{t+1}(S_{t+1}))$ for each state as survivor state trajectories; and

selecting $\hat{I}_t(S_t)$s that correspond to the selected state trajectories as $I_{t,p+1}(S_t)$.

9. The method of claim 8, further comprising:

a. assigning $L(S_T)=0$ for all states at $t = T$;

b. generating $m(\hat{I}_t(S_t))$ for all state transitions between states S_t and all possible states S_{t+1};

c. selecting state transitions between the states S_t and S_{t+1} that have a largest $L(S_t) = L(S_{t+1}) + m(\hat{I}_{t+1}(S_{t+1}))$ and $\hat{I}_{t+1}(S_{t+1})$ that correspond to the selected state transitions;

d. updating the survivor state trajectories at states S_t by adding the selected state transitions to the corresponding survivor state trajectories at state S_{t+1};

e. decrementing t by 1;

f. repeating b-e until $t = 0$; and

g. selecting all the $\hat{I}_t(S_t)$ that correspond to a survivor state trajectory that corresponding to a largest $L(S_t)$ at $t = 0$ as $I_{1,p+1}^T$.

10. The method of claim 6, wherein the channel is modeled as $\mathbf{P}_c(Y \mid X) = \mathbf{P}_c\mathbf{B}_c(Y \mid X)$ where $\mathbf{P}_c$ is a channel state transition probability matrix and $\mathbf{B}_c(Y \mid X)$ is a

diagonal matrix of state output probabilities, the method comprising for each t = 1, 2, ..., T:

generating $\gamma_{t,i}(I_{1,p}^T) = \alpha_i(Y_1^t \mid I_{1,p}^t)\beta_i(Y_{t+1}^T \mid I_{t+1,p}^T)$;

selecting an $\hat{I}_t(S_t)$ that maximizes $L(S_t)=L(S_{t+1})+ m(\hat{I}_{t+1}(S_{t+1}))$, where $m(\hat{I}_t(S_t))$ is defined as

$m(\hat{I}_t(S_t)) = \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T)\beta_j(Y_t \mid X_t(S_t))$, n_c being a number of states in an HMM of the channel;

selecting state transitions between states S_t and S_{t+1} that corresponds to a largest $L(S_t) = L(S_{t+1}) + m(\hat{I}_{t+1}(S_{t+1}))$; and

forming survivor state trajectories by connecting selected state transitions.

11. The method of claim 10, further comprising:

selecting $\hat{I}_t(S_t)$ that corresponds to a survivor state trajectory at t = 0 that has the largest $L(S_t)$ as $I_{1,p+1}^T$ for each pth iteration;

comparing $I_{1,p}^T$ and $I_{1,p+1}^T$; and

outputting $I_{1,p+1}^T$ as the second decode result if $I_{1,p}^T$ and $I_{1,p+1}^T$ are within the compare threshold.

12. A maximum a posteriori (MAP) decoder that decodes a transmitted information sequence using a received information sequence received through a channel, comprising:

a memory; and

a controller coupled to the memory, the controller iteratively generating a sequence of one or more decode results starting with an initial decode result, and outputting one of adjacent decode results as a decode of the input information sequence if the adjacent decode results are within a compare threshold.

13. The decoder of claim 12, wherein the controller:

a. generates the initial decode result as a first decode result;

b. generates a second decode result based on the first decode result and a model of the channel;

c. compares the first and second decode results;

d. replaces the first decode result with the second decode result; and

e. repeats b-d until the first and second decode result are not within the compare threshold.

14. The decoder of claim 13, wherein the controller searches for information sequence that maximizes a value of an auxiliary function.

15. The decoder of claim 14, wherein the auxiliary function is based on expectation maximization (EM).

16. The decoder of claim 15, wherein the model of the channel is a Hidden Markov Model (HMM) having an initial state probability vector π and probability density matrix (PDM) of $\mathbf{P}(X,Y)$, where $X \in \mathbf{X}$, $Y \in \mathbf{Y}$ and elements of $\mathbf{P}(X,Y)$, $p_{ij}(X,Y) = \Pr(j,X,Y \mid i)$, are conditional probability density functions of an information element X of the second information sequence that corresponds to a received element Y of the first information sequence after the HMM transfers from a state i to a state j, the auxiliary function being expressed as:

$Q(X_1^T, X_{1,p}^T) = \sum_z \Psi(z, X_{1,p}^T, Y_1^T)\log\left(\Psi(z, X_1^T, Y_1^T)\right)$, where p is a number of iterations, $\Psi(z,\ X_1^T,\ Y_1^T) = \pi_{i_0}\prod_{t=1}^{T} p_{i_{t-1}i_t}(X_t, Y_t)$, T is a number of information elements in a particular information sequence, z is a HMM state sequence i_0^T, π_{i_0} is the probability of an initial state i_0, X_1^T is the second information sequence, $X_{1,p}^T$ is a second information sequence estimate corresponding to a pth iteration, and Y_1^T is the first information sequence.

17. The decoder of claim 16, wherein the auxiliary function is expanded to be:

$$Q(X_1^T, X_{1,p}^T) = \sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{j=1}^{n} \gamma_{t,ij}(X_{1,p}^T)\log\left(p_{ij}(X_t, Y_t)\right) + C$$

where C does not depend on X_1^T and

$$\gamma_{t,ij}(X_{1,p}^T) = \alpha_i(X_{1,p}^{t-1}, Y_1^{t-1})\,p_{ij}(X_{t,p}, Y_t)\,\beta_j(X_{t+1,p}^T, Y_{t+1}^T)$$

where $\alpha_i(X_{1,p}^t, Y_1^T)$ and $\beta_j(X_{t+1,p}^T, Y_{t+1}^T)$ are the elements of forward and backward probability vectors defined as

$$\alpha(X_1^t,\ Y_1^t) = \pi\prod_{i=1}^{t} P(X_i, Y_i),\ \text{and}\ \beta(X_{t+1}^T, Y_{t+1}^T) = \prod_{j=t+1}^{T} P(X_j, Y_j)\,\mathbf{1},\ \pi \text{ is an initial}$$

probability vector, **1** is the column vector of ones.

18. The decoder of claim 17, wherein a source of an encoded sequence is a trellis code modulator (TCM), the TCM receiving a source information sequence I_1^T and outputting X_1^T as an encoded information sequence that is transmitted, the TCM defining $X_t = g_t(S_t, I_t)$ where X_t and I_t are the elements of X_1^T and I_1^T for each time t, respectively, S_t is a state of the TCM at t, and $g_t(.)$ is a function relating X_t, to I_t and S_t, the controller generates, for iteration p+1, an input information sequence estimate $I_{1,p+1}^T$ that corresponds to a sequence of TCM state transitions that has a longest cumulative distance $L(S_{t-1})$ at t = 1 or $L(S_0)$, wherein a distance for each of the TCM state transitions is defined by $L(S_{t+1}) = L(S_{t+1}) + m(\hat{I}_{t+1}(S_{t+1}))$ for the TCM state transitions at each t for t = 1, ..., T and the cumulative distance is the sum of $m(\hat{I}_t(S_t))$ for all t, $m(\hat{I}_t(S_t))$ being defined as

$$m\left(\hat{I}_t(S_t)\right) = \sum_{i=1}^{n_c}\sum_{j=1}^{n_c} \gamma_{t,ij}\left(I_{1,p}^T\right)\log p_{c,ij}\left(Y_t \mid X_t(S_t)\right),\ \text{for each } t = 1, 2, \ldots, T,\ \text{where}$$

$X_t(S_t) = g_t(S_t, \hat{I}_t(S_t))$, n_c is a number of states in an HMM of the channel and $p_{c,ij}(Y_t \mid X_t(S_t))$ are channel conditional probability density functions of Y_t when $X_t(S_t)$ is transmitted by the TCM, $I_{1,p+1}^T$ being set to a sequence of $\hat{I}_t$ for all t.

19. The decoder of claim 18, wherein for each t = 1, 2, ..., T, the controller generating $m(\hat{I}_t(S_t))$ for each possible state transition of the TCM, selecting state trajectories that correspond to largest $L(S_t) = L(S_{t+1}) + m(\hat{I}_{t+1}(S_{t+1}))$ for each state as survivor state trajectories, and selecting $\hat{I}_{t+1}(S_{t+1})$s that correspond to the selected state trajectories as $I_{t+1,p+1}(S_{t+1})$.

20. The decoder of claim 19, wherein the controller:

a. assigns $L(S_T)=0$ for all states at t = T;

b. generates $m(\hat{I}_t(S_t))$ for all state transitions between states S_t and all possible states S_{t+1};

c. selects state transitions between the states S_t and S_{t+1} that have a largest $L(S_t) = L(S_{t+1}) + m(\hat{I}_{t+1}(S_{t+1}))$ and $\hat{I}_{t+1}(S_{t+1})$ that correspond to the selected state transitions;

d. updates the survivor state trajectories at states S_t by adding the selected state transitions to the corresponding survivor state trajectories at state S_{t+1};

e. decrements t by 1;

f. repeats b-e until $t = 0$; and

g. selects all the $\hat{I}_t(S_t)$ that correspond to a survivor state trajectory that corresponding to a largest $L(S_t)$ at $t = 0$ as $I_{1,p+1}^T$.

21. The decoder of claim 20, wherein the channel is modeled as $\mathbf{P}_c(Y \mid X) = \mathbf{P}_c\mathbf{B}_c(Y \mid X)$ where $\mathbf{P}_c$ is a channel state transition probability matrix and $\mathbf{B}_c(Y \mid X)$ is a diagonal matrix of state output probabilities, for each $t = 1, 2, \ldots, T$, the controller:

generates $\gamma_{t,i}(I_{1,p}^T) = \alpha_i(Y_1^t \mid I_{1,p}^t)\beta_i(Y_{t+1}^T \mid I_{t+1,p}^T)$;

selects an $\hat{I}_t(S_t)$ that maximizes $L(S_t)=L(S_{t+1})+ m(\hat{I}_{t+1}(S_{t+1}))$, where $m(\hat{I}_t(S_t))$ is defined as

$$m(\hat{I}_t(S_t)) = \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T)\beta_j(Y_t \mid X_t(S_t)),$$ n_c being a number of states in an HMM of the channel;

selects state transitions between states S_t and S_{t+1} that corresponds to a largest $L(S_t) = L(S_{t+1}) + m(\hat{I}_{t+1}(S_{t+1}))$; and

forms survivor state trajectories by connecting selected state transitions.

22. The decoder of claim 21, wherein the controller selects $\hat{I}_t(S_t)$ that corresponds to a survivor state trajectory at $t = 0$ that has the largest $L(S_t)$ as $I_{1,p+1}^T$ for each pth iteration, compares $I_{1,p}^T$ and $I_{1,p+1}^T$, and outputs $I_{1,p+1}^T$ as the second decode result if $I_{1,p}^T$ and $I_{1,p+1}^T$ are within the compare threshold.

ABSTRACT OF THE DISCLOSURE

This invention provides an iterative process to maximum a posteriori (MAP) decoding. The iterative process uses an auxiliary function which is defined in terms of a complete data probability distribution. The auxiliary function is derived based on an expectation maximization (EM) algorithm. For a special case of trellis coded modulators, the auxiliary function may be iteratively evaluated by a combination of forward-backward and Viterbi algorithms. The iterative process converges monotonically and thus improves the performance of any decoding algorithm. The MAP decoding minimizes a probability of error. A direct approach to achieve this minimization results in complexity which grows exponentially with T, where T is the size of the input. The iterative process avoids this complexity by converging on the MAP solution through repeated maximization of the auxiliary function.

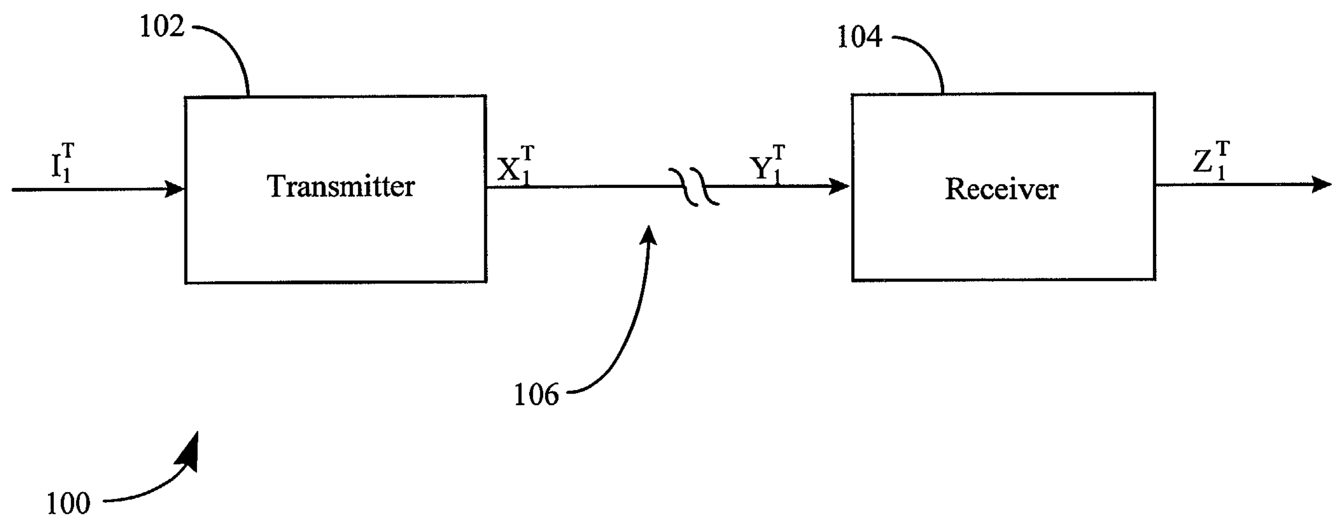


Fig. 1

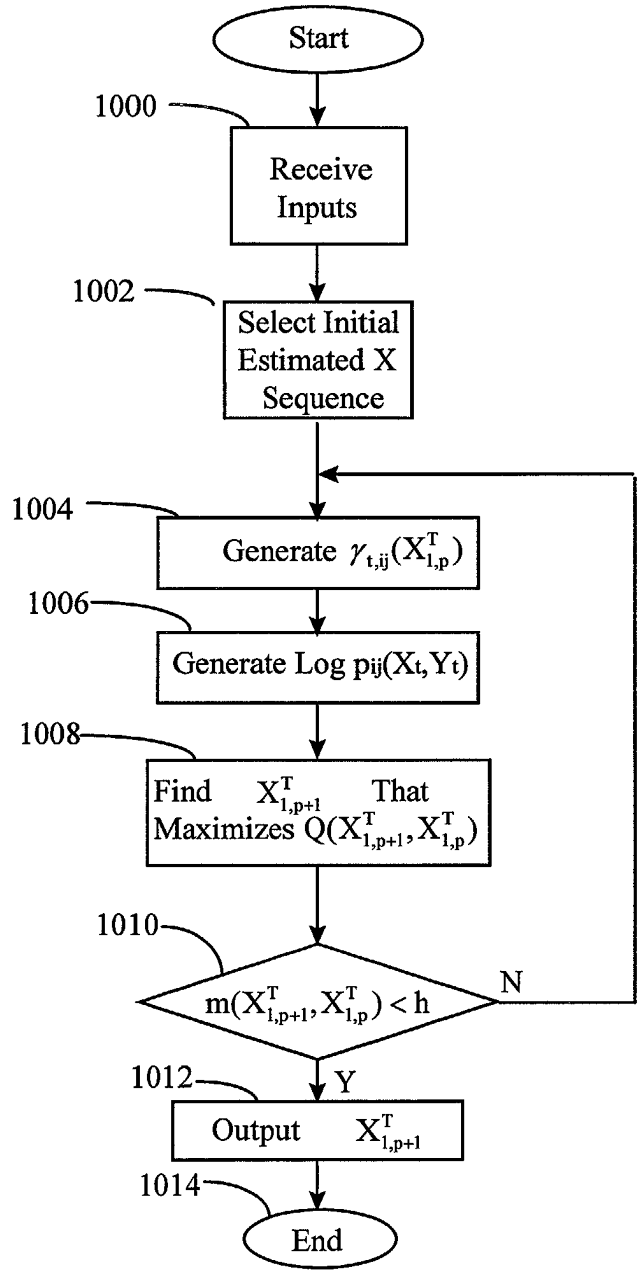

Start
1000
Receive Inputs
1002
Select Initial Estimated X Sequence
1004
Generate $\gamma_{t,ij}(X^T_{1,p})$
1006
Generate Log pij(Xt, Yt)
1008
Find $X^T_{1,p+1}$ That Maximizes $Q(X^T_{1,p+1}, X^T_{1,p})$
1010
$m(X^T_{1,p+1}, X^T_{1,p}) < h$
N
Y
1012
Output $X^T_{1,p+1}$
1014
End

Fig. 2

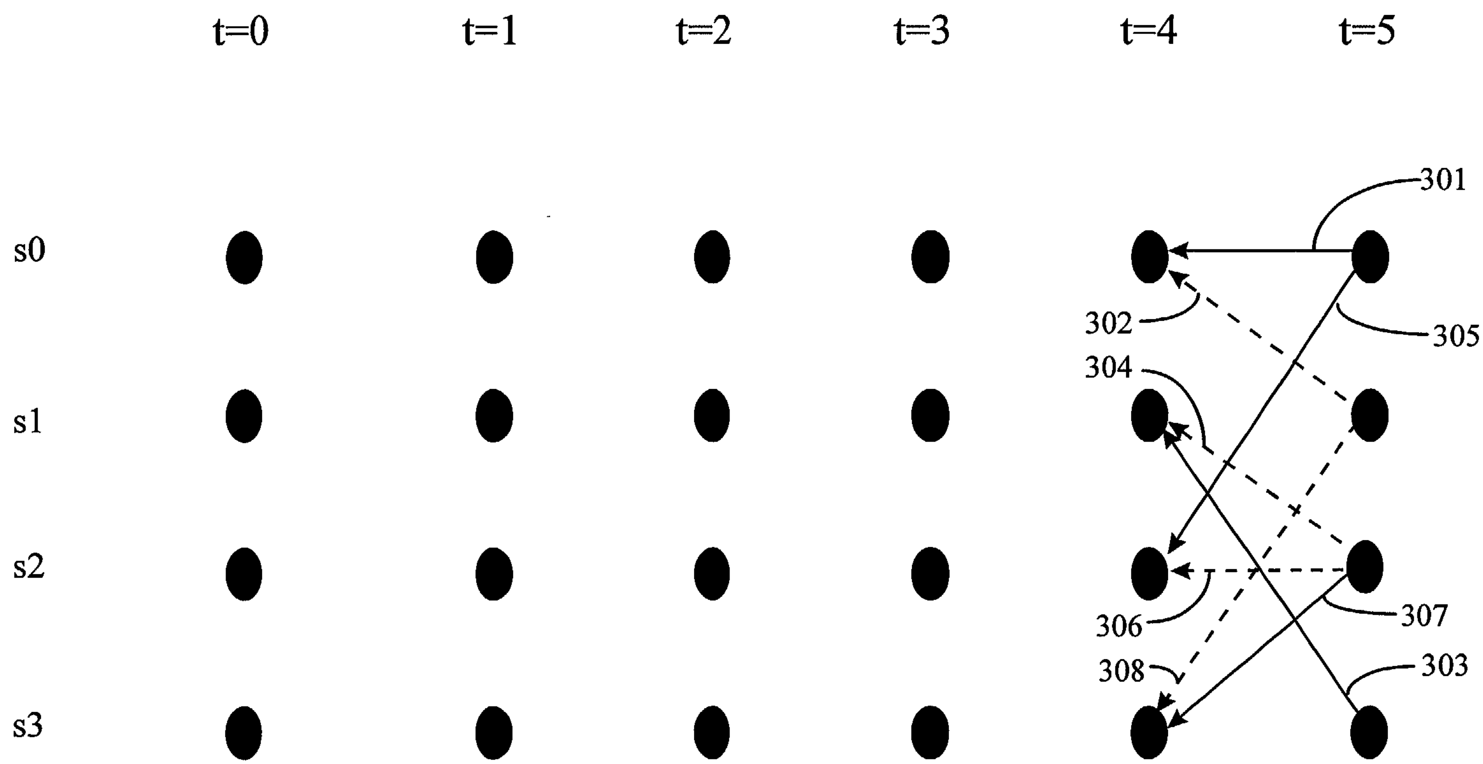
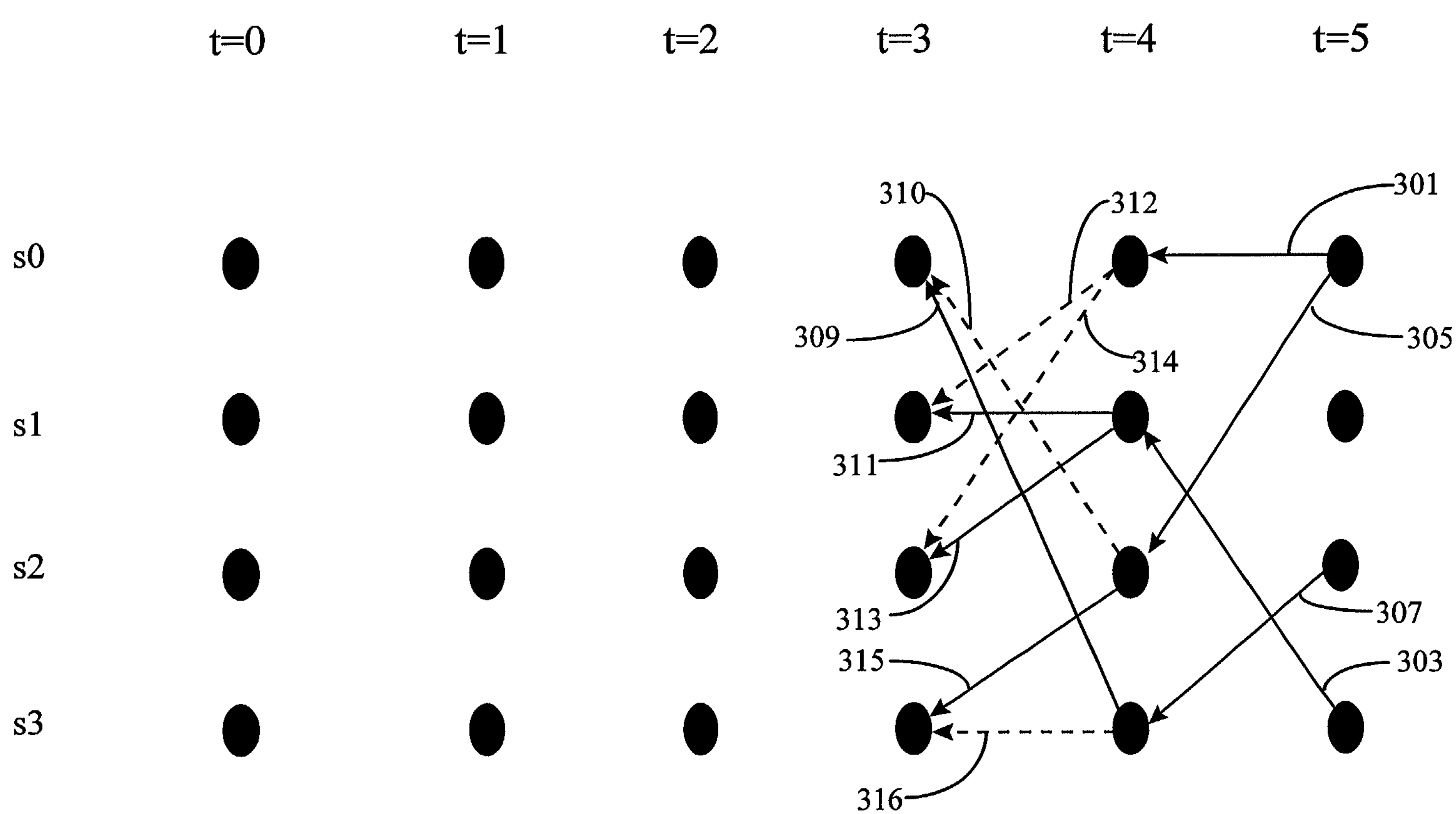


Fig. 3



t=0
t=1
t=2
t=3
t=4
t=5
s0
s1
s2
s3
301
305
307
303
312
314
310
309
311
313
315
316

Fig. 4

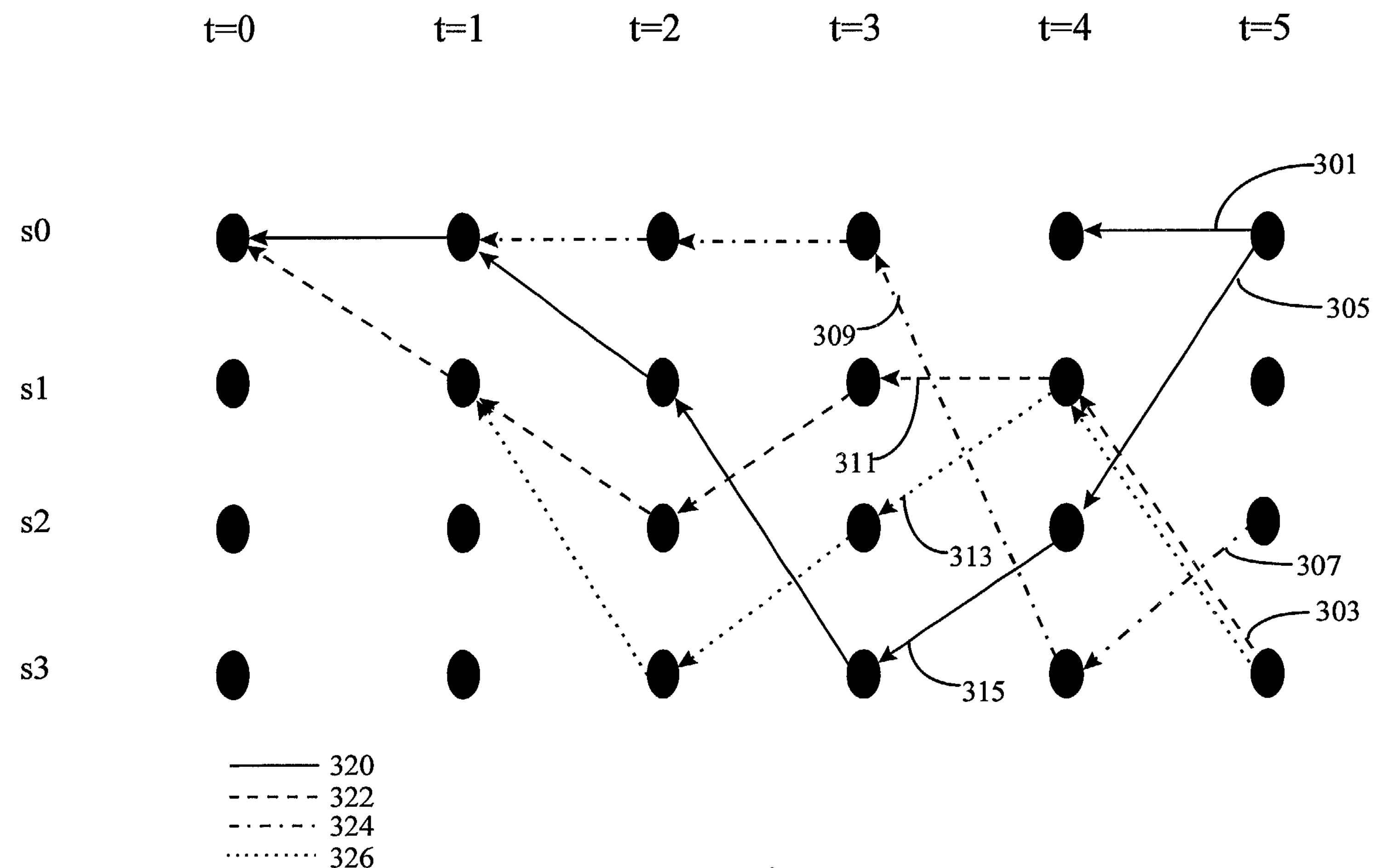

t=0
t=1
t=2
t=3
t=4
t=5
s0
s1
s2
s3
301
305
309
311
313
307
303
315
320
322
324
326

Fig. 5

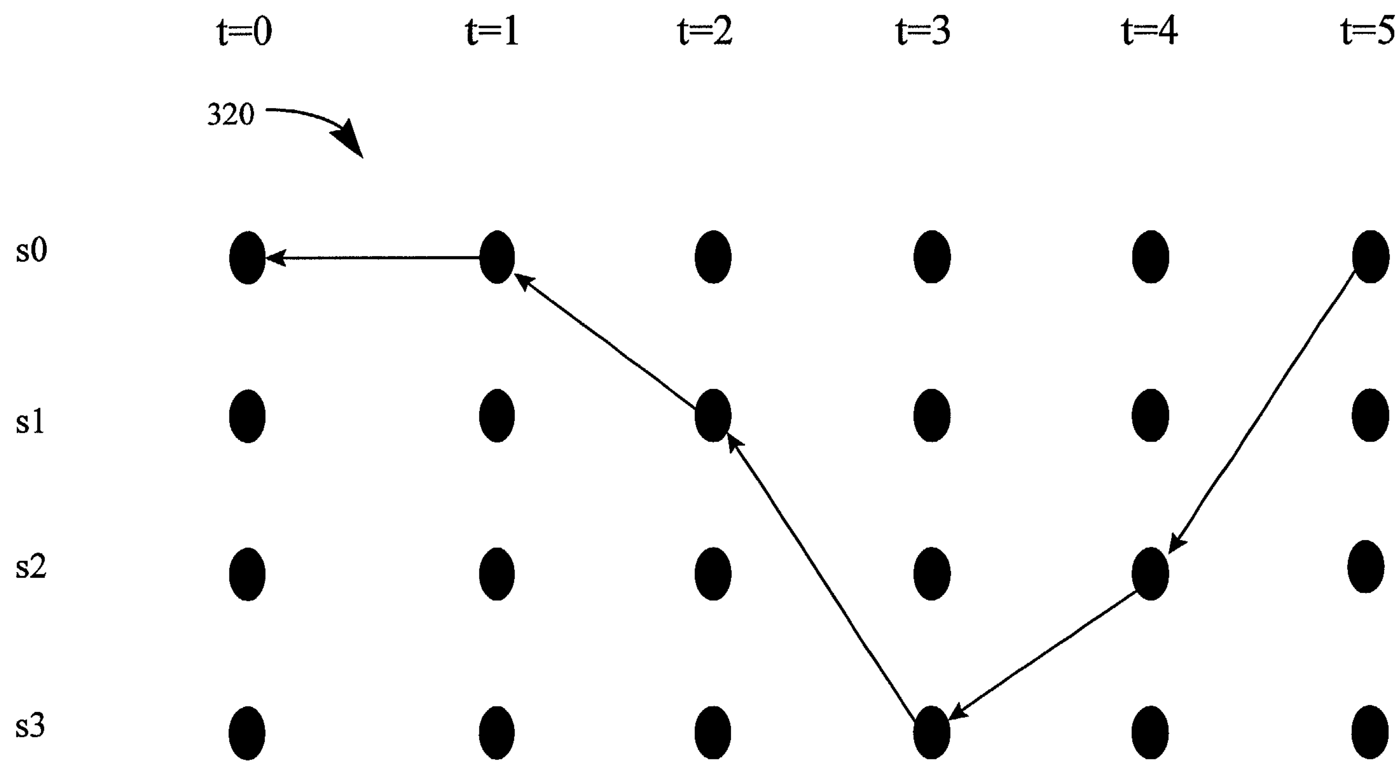
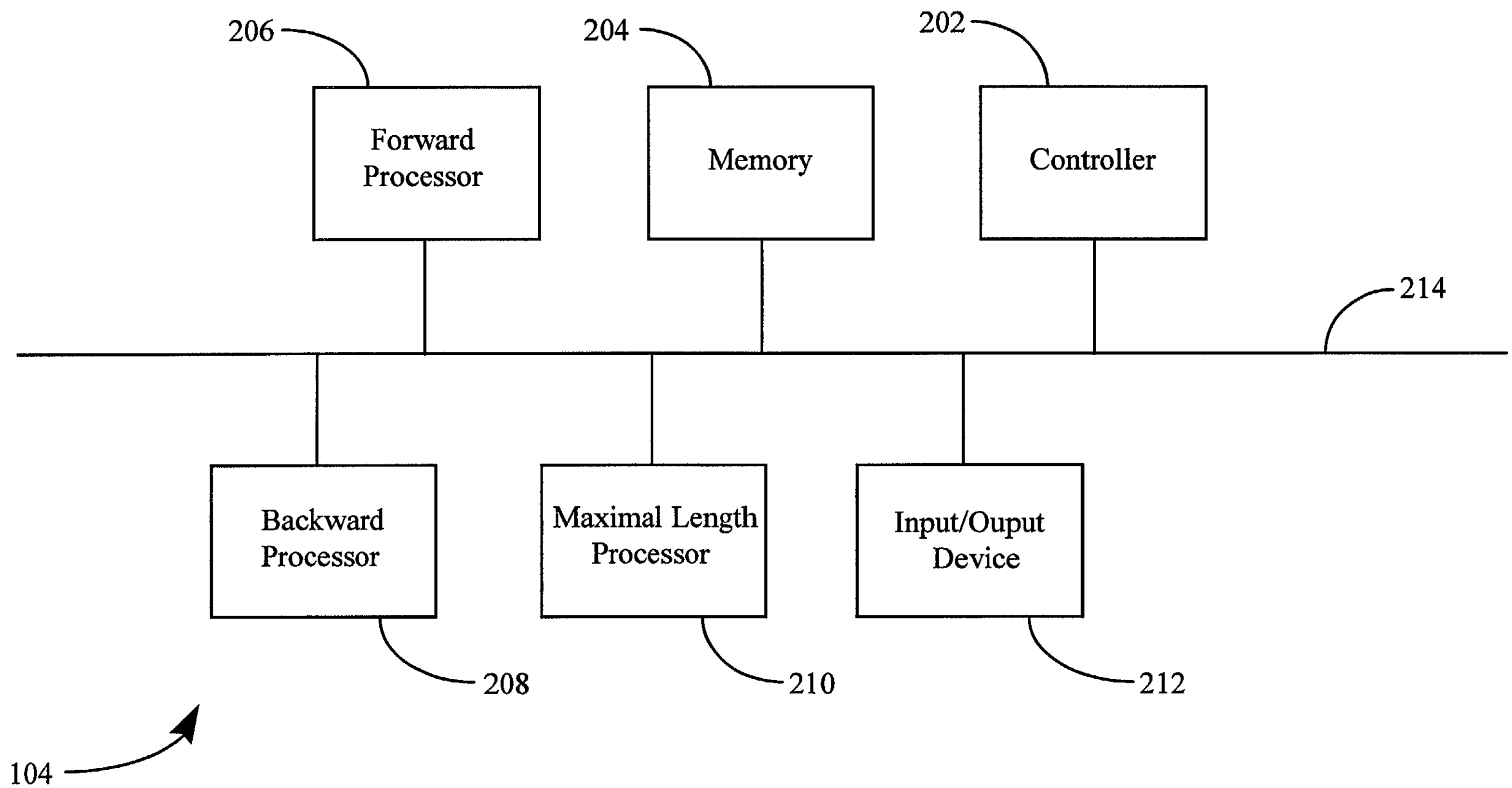


Fig. 6



206
Forward Processor
204
Memory
202
Controller
214
Backward Processor
Maximal Length Processor
Input/Ouput Device
208
210
212
104

Fig. 7

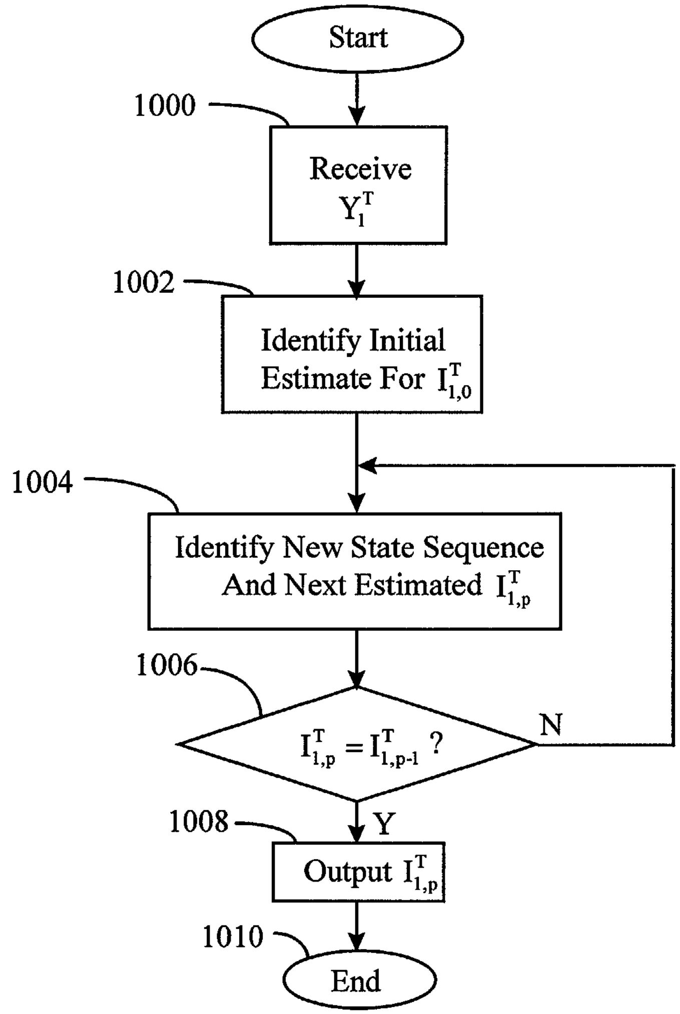

Start
1000
Receive Y_1^T
1002
Identify Initial Estimate For $I_{1,0}^T$
1004
Identify New State Sequence And Next Estimated $I_{1,p}^T$
1006
$I_{1,p}^T = I_{1,p-1}^T$?
N
Y
1008
Output $I_{1,p}^T$
1010
End

Fig. 8

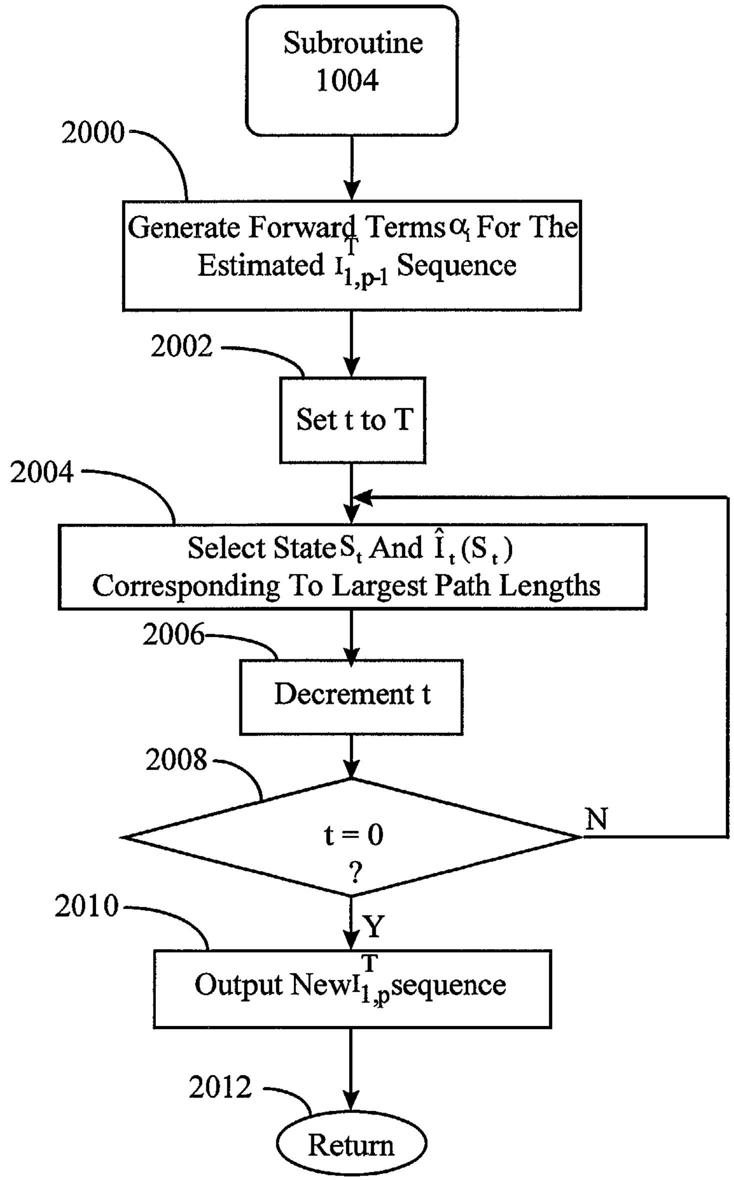

Subroutine
1004
2000
Generate Forward Terms α_t For The Estimated $I^T_{1,p-1}$ Sequence
2002
Set t to T
2004
Select State S_t And $\hat{I}_t(S_t)$ Corresponding To Largest Path Lengths
2006
Decrement t
2008
t = 0
?
N
Y
2010
Output New $I^T_{1,p}$ sequence
2012
Return

Fig. 9

Docket No. 1999-0078

IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

Declaration and Power of Attorney

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am an original, first and sole inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled **ITERATIVE DECODING**, the specification of which is attached hereto.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by an amendment, if any, specifically referred to in this oath or declaration.

I acknowledge the duty to disclose all information known to me which is material to patentability as defined in Title 37, Code of Federal Regulations, 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, 119 of any foreign application(s) for patent or inventors' certificate and/or United States provisional application(s) filed by me or my legal representatives or assigns within one year prior to the application as listed below and have also identified below any foreign application for patent or inventors' certificate and/or United States provisional application(s) having a filing date before that of the application on which priority is claimed:

Serial No.: 60/174,601
Filed: January 5, 2000

I hereby claim the benefit under Title 35, United States Code, 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, Unites States Code, 112, I acknowledge the duty to disclose all information known to us to be material to patentability as defined in Title 37, Code of Federal Regulations, 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

Serial No.
Filed
Status:

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

I hereby appoint the following attorney(s) with full power of substitution and revocation, to prosecute said application, to make alterations and amendments therein, to receive the patent, and to transact all business in the Patent and Trademark Office connected therewith:

Samuel H. Dworetsky	(Reg. No. 27,873)
Thomas A. Restaino	(Reg. No. 33,444)
Michele L. Conover	(Reg. No. 34,962)
Cedric G. de la Cruz	(Reg. No. 36,498)
Benjamin S. Lee	(Reg. No. 42,787)
Robert B. Levy	(Reg. No. 28,234)
Alfred G. Steinmetz	(Reg. No. 22,971)

I also appoint James A. Oliff (Reg. No. 27,075), William P. Berridge (Reg. No. 30,024), Kirk M. Hudson (Reg. No. 27,562), Thomas J. Pardini (Reg. No. 30,411), Edward P. Walker (Reg. No. 31,450), Robert A. Miller (Reg. No. 32,771), Mario A. Costantino (Reg. No. 33,565), and Stephen J. Roe (Reg. No. 34,463) of OLIFF & BERRIDGE, PLC as associate attorneys, with full power to prosecute said application, to make alterations and amendments therein, and to transact all business in the U.S. Patent and Trademark Office connected therewith.

Please address all correspondence to S. H. Dworetsky, AT&T Corp., P.O. Box 4110, Middletown, New Jersey 07748. However, telephone calls should be made to Benjamin Lee at (973) 360-8117.

Full name of inventor: William Turin

Inventor's signature William Turin Date April 11, 2000

Residence: East Brunswick, Middlesex County, New Jersey

Citizenship: United States

Post Office Address: 25 Hillwood Road
East Brunswick, New Jersey 08816

Appendix

MAP Decoding in Channels with Memory

William Turin, *Senior Member, IEEE*

Abstract

The expectation-maximization (EM) algorithm is popular in estimating parameters of various statistical models. In this paper, we consider applications of the EM algorithm to the maximum a posteriori (MAP) sequence decoding assuming that sources and channels are described by hidden Markov models (HMMs). HMMs can accurately approximate a large variety of communication channels with memory and, in particular, wireless fading channels with noise. The direct maximization of the a posteriori probability (APP) is too complex. The EM algorithm allows us to obtain the MAP sequence estimation iteratively. Since each step of the EM algorithm increases the APP, the algorithm can improve performance of any decoding procedure.

Keywords: Maximum a posteriori decoding, EM algorithm, hidden Markov model, fading channel

W. Turin is with the Communications Research Department, AT&T Labs-Research, 180 Park Avenue, Florham Park, NJ 07932 (e-mail: wt@research.att.com).

1. INTRODUCTION

Maximum a posteriori (MAP) sequence decoding is optimum, because it minimizes the probability of error. [27] It is usually performed by the Viterbi algorithm and most recently by the turbo decoding algorithm for a special class of codes. These algorithms are directly applicable if communications channels are memoryless. However, they perform an approximate MAP decoding if channel errors are bursty which is the case in wireless communications due to fading. It is usually very difficult to find a sequence MAP estimate directly for channels with memory. The expectation-maximization (EM) algorithm has been successfully applied by Georghiades and Han, [8] Zeger and Kobayashi, [28] and Georghiades [9] to decoding information transmitted over the fading wireless channel. In these papers, the fading channel is modeled by a complex Gaussian process.

Alternatively, the wireless channel can be modeled by hidden Markov models (HMMs). [22,23] It can be shown that HMMs are general enough to approximate not only fading, but also other types of signal distortion such as interference and non-Gaussian noise. [23] In this paper, we demonstrate that the EM algorithm can be applied to MAP decoding if channel errors are described by an HMM. In our approach, the signal parameters are obtained by applying the EM algorithm to maximizing the a posteriori probability (APP) of the transmitted symbols.

In developing the MAP decoding algorithm it is usually assumed that the communication channel is memoryless which is achieved by interleaving. However, in the majority of real channels the error dependence extends over long time intervals. Complete independence is impossible to achieve due to the information delivery delay constraints and system memory limitations. We should also remember that a memoryless channel has lower capacity than a channel with memory with the same bit-error rate. [7,10] Therefore, it is important to develop decoding algorithms for channels with memory.

There are many different models of channels with memory which correspond to various channel impairments such as fading, interference, and noise. All of them can be accurately approximated by hidden Markov models (HMMs). It can be shown that HMMs represent a dense family allowing us to approximate a large variety of stochastic processes. Their application in

many diverse fields (such as speech, image, and handwriting recognition, experimental genetics, sociology, stock market modeling, etc) serves as experimental evidence of their generality. Another reason for their popularity is the relative simplicity of their use.

In this paper we develop the MAP decoding algorithm for a general input-output HMM (IOHMM) which, as we will see, incorporates source and channel HMMs. For the special class of models, the decoding can be performed using the Viterbi algorithm. However, in the general case this algorithm gives only an approximate solution. We demonstrate that this approximate solution can be improved by the expectation maximization (EM) algorithm. The paper is organized as follows. In section 2, we discuss the relationship between various decoding criteria. In section 3, we consider the IOHMMs and their application to computing the transmitted symbol sequence APP. In section 4, we develop the MAP EM decoding algorithm and illustrate its application to decoding of block and convolutional codes.

2. MAP DECODING

Suppose that we have an input-output system whose input is described by a sequence of symbols $\boldsymbol{X}_1^T = (X_1, X_2, \ldots, X_T)$ and the corresponding output is $\boldsymbol{Y}_1^T = (Y_1, Y_2, \ldots, Y_T)$. Our goal is to choose the most probable input which produced the observed output $\boldsymbol{Y}_1^T$.

The optimal estimator that maximizes the probability of $\boldsymbol{X}_1^T$ correct decoding is the maximum a posteriori (MAP) estimator: [27]

$$\hat{\boldsymbol{X}}_1^T = \underset{\boldsymbol{X}_1^T}{argmax}\, Pr(\boldsymbol{X}_1^T \mid \boldsymbol{Y}_1^T) \tag{1}$$

where $Pr(\cdot)$ denotes the corresponding probability or probability density function. Since the maximization does not depend on $\boldsymbol{Y}_1^T$, the MAP estimate can be obtained by maximizing the unnormalized APP

$$\hat{\boldsymbol{X}}_1^T = \underset{\boldsymbol{X}_1^T}{argmax}\, Pr(\boldsymbol{X}_1^T, \boldsymbol{Y}_1^T) = \underset{\boldsymbol{X}_1^T}{argmax}\, Pr(\boldsymbol{Y}_1^T \mid \boldsymbol{X}_1^T) Pr(\boldsymbol{X}_1^T). \tag{2}$$

It follows from this equation that the maximum likelihood (ML) and MAP estimates coincide if the input is uniformly distributed. This assumption is often made when there is no information about the input probability distribution. [27] However, better results can be achieved if we

exploit the input (source) statistics. Since the ML estimate can be viewed as a special case of the MAP estimate, we consider only the latter in the sequel.

It is often necessary to estimate a subset of the input sequence and in particular a single symbol:

$$\hat{X}_t = \underset{X_t}{argmax}\, Pr(X_t, Y_1^T). \tag{3}$$

In many applications the estimation must be performed before receiving the whole sequence Y_1^T. If estimation of X_t is made on the basis of $Y_1^{t+\tau}$ with $\tau > 0$, it is called a *fixed lag smoothing*; if $\tau = 0$, it is called a *filtering*; if $\tau < 0$, it is called a *prediction*. To solve the previous equations we need to develop algorithms for calculating the corresponding probability measures and their maximization. This is done by modeling the input-output system.

3. HIDDEN MARKOV MODELS

An input-output HMM $\lambda = (S, X, Y, \boldsymbol{\pi}, \{\mathbf{P}(x,y)\})$ is defined by its internal states $S = \{1,2,\ldots,n\}$, inputs X, outputs Y, initial state probability vector $\boldsymbol{\pi}$, and the input-output probability density matrices (PDM) $\mathbf{P}(x,y)$, $x \in X$, $y \in Y$ whose elements $p_{ij}(x,y) = Pr(j,x,y \mid i)$ are the conditional probability density functions (PDF) of input x and corresponding output y after transferring from state i to state j. It is assumed that the state sequence $\boldsymbol{S}_0^t = (S_0, S_1, \ldots, S_t)$, input sequence $\boldsymbol{X}_1^t = (X_1, X_2, \ldots, X_t)$, and output sequence $\boldsymbol{Y}_1^t = (Y_1, Y_2, \ldots, Y_t)$ possess the following Markovian property

$$Pr(S_t, X_t, Y_t \mid \boldsymbol{S}_0^{t-1}, \boldsymbol{X}_1^{t-1}, \boldsymbol{Y}_1^{t-1}) = Pr(S_t, X_t, Y_t \mid S_{t-1}).$$

According to this model, the PDF of the input sequence $\boldsymbol{X}_1^T$ and output sequence $\boldsymbol{Y}_1^T$ has the form [23]

$$p_T(\boldsymbol{X}_1^T, \boldsymbol{Y}_1^T) = \boldsymbol{\pi} \prod_{i=1}^{T} \mathbf{P}(X_i, Y_i)\, \mathbf{1} \tag{4}$$

where $\mathbf{1}$ is a column vector of n ones.

If the source sequence is modeled by an autonomous HMM with states $S_t^{(s)}$ and PDM $\mathbf{P}_s(x) = [Pr(S_t^{(s)}, x \mid S_{t-1}^{(s)})]_{n_s, n_s}$ and, for each input sequence, the channel is modeled by the

conditional HMM with states $S_t^{(c)}$ and PDM $\mathbf{P}_c(y|x) = [Pr(S_t^{(c)}, y | S_{t-1}^{(c)}, x)]_{n_c,n_c}$, then the combined IOHMM is described by the PDM of the form

$$\mathbf{P}(X,Y) = \mathbf{P}_s(X) \otimes \mathbf{P}_c(Y|X) \tag{5}$$

where $\otimes$ denotes the matrix Kronecker product. In other words, the combined model is an HMM whose states represent all possible combinations of the transmitted sequence states and channel states.

In the most popular model, the conditional output PDM has the form

$$\mathbf{P}_c(Y|X) = \mathbf{P}_c \mathbf{B}_c(Y|X) \tag{6}$$

where $\mathbf{P}_c = [p_{ij}]_{n_c,n_c}$ is the channel state transition probability matrix and $\mathbf{B}_c(Y|X)$ is a diagonal matrix of the state output probabilities. For example, according to the Gilbert-Elliott [3,10] model, the channel has two states: "good" and "bad". In the good state, errors occur with small probability b_1 while in the bad state they occur with larger probability b_2. If we assume that the first state is good and the second state is bad, then the model is described by equation (6) with

$$\mathbf{B}_c(X|X) = \begin{bmatrix} 1-b_1 & 0 \\ 0 & 1-b_2 \end{bmatrix} \quad \mathbf{B}_c(\bar{X}|X) = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix} \tag{7}$$

where $\bar{X}$ denotes the complement of X. The two-state model is the simplest HMM for channels with memory. Models with larger state space are often needed. [19,23] There are several HMMs for fading channels with additive white Gaussian noise (AWGN). In these models, the HMM states are usually associated with the channel fade levels while the state output conditional probability density functions are Gaussian [13,18,22,23]

$$b_k(Y_t|X_t) \propto \exp(-\parallel Y_t - a_k X_t \parallel^2 / N_0). \tag{8}$$

Let us consider now the transmitted sequence modeling by HMMs. Usually, the source HMM is obtained by fitting it to experimental data. This process is called the HMM training. [16] If the transmitted sequence is generated by a trellis coded modulator (TCM), [6,11,24] then it can be described as a finite state machine:

$$S_{t+1}^{(s)} = f_t(S_t^{(s)}, I_t)$$
$$X_t = g_t(S_t^{(s)}, I_t). \tag{9}$$

After receiving an information symbol I_t in state $S_t^{(s)}$ the machine transfers to state $S_{t+1}^{(s)} = f_t(S_t^{(s)}, I_t)$ and outputs a symbol $X_t = g_t(S_t^{(s)}, I_t)$. The system is called TCM, because its state transition graph resembles a trellis whose nodes represent the states $S_t^{(s)}$ for $t = 1, 2, \ldots, T$ and edges represent all possible state transitions. In a typical implementation, the first equation in (9) represents a convolutional encoder while the second equation represents a modulator.

It follows from this description that the modulator output symbols are not independent, even if the source symbols are independent. From a statistical point of view, TCM can be described as a method of creating correlated sequences that are resistant to channel errors, allowing us to recover the source sequences with high reliability. It follows from the TCM definition that if the source symbols can be modeled by an HMM, the output process is also an HMM.

4. MAP SEQUENCE DECODING VIA THE EM ALGORITHM

If the channel is modeled by an IOHMM, the MAP estimate of the sequence X_1^T maximizes the right hand side of equation (4). However, as we can see, the direct maximization is a difficult problem. If X_t is a discrete variable, then we need to consider all possible sequences X_1^T. Thus, the complexity grows exponentially with T. In the special case, when the input sequence X_1^T is uniquely determined by the sequence of states S_1^T, the maximum can be found by the Viterbi algorithm. [4] In this paper, we show that the general problem can be solved iteratively using the EM algorithm which, in our case, is a combination of the forward-backward and Viterbi algorithms. [2,16] The EM algorithm converges monotonically, [2] therefore, even single step of the EM algorithm allows us to improve performance of any decoding algorithm.

To develop the algorithm, we define the complete data probability distribution as [23, Sec. 3.2.2]

$$\psi(z, X_1^T, Y_1^T) = \pi_{i_0} \prod_{t=1}^{T} p_{i_{t-1} i_t}(X_t, Y_t) \tag{10}$$

where $z = i_0^T$ is the HMM state sequence and $p_{ij}(X, Y)$ are the elements of the matrix $\mathbf{P}(X, Y)$. The MAP sequence estimate of equation (2) can be obtained iteratively by the following EM

algorithm [2] [23]

$$X_{1,p+1}^T = \arg\max_{X_1^T} Q(X_1^T, X_{1,p}^T), \quad p=0,1,2,... \tag{11}$$

where $Q(X_1^T, X_{1,p}^T)$ is the auxiliary function which, in our case, can be written as

$$Q(X_1^T, X_{1,p}^T) = \sum_z \psi(z, X_{1,p}^T, Y_1^T) \log \psi(z, X_1^T, Y_1^T).$$

Using the relationship between equations (10) and (4) we can rewrite the auxiliary function in the following form [23]

$$Q(X_1^T, X_{1,p}^T) = \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{t,ij}(X_{1,p}^T) \log p_{ij}(X_t, Y_t) + C \tag{12}$$

where C does not depend on X_1^T

$$\gamma_{t,ij}(X_{1,p}^T) = \alpha_i(X_{1,p}^{t-1}, Y_1^{t-1}) p_{ij}(X_{t,p}, Y_t) \beta_j(X_{t+1,p}^T, Y_{t+1}^T)$$

$\alpha_i(X_{1,p}^t, Y_1^t)$ and $\beta_i(X_{t+1,p}^T, Y_{t+1}^T)$ are the elements of the following forward and backward probability vectors

$$\boldsymbol{\alpha}(X_1^t, Y_1^t) = \boldsymbol{\pi} \prod_{i=1}^{t} \mathbf{P}(X_i, Y_i) \quad \text{and} \quad \boldsymbol{\beta}(X_1^t, Y_1^t) = \prod_{i=t}^{T} \mathbf{P}(X_i, Y_i)\mathbf{1}. \tag{13}$$

It follows from these equations that the forward probability vectors can be evaluated by the forward algorithm

$$\boldsymbol{\alpha}(X_1^0, Y_1^0) = \boldsymbol{\pi}, \quad \boldsymbol{\alpha}(X_1^t, Y_1^t) = \boldsymbol{\alpha}(X_1^{t-1}, Y_1^{t-1}) \mathbf{P}(X_t, Y_t) \tag{14}$$

and the backward probability vectors can be evaluated by the backward algorithm

$$\boldsymbol{\beta}(X_{T+1}^T, Y_{T+1}^T) = \mathbf{1}, \quad \boldsymbol{\beta}(X_t^T, Y_t^T) = \mathbf{P}(X_t, Y_t) \boldsymbol{\beta}(X_{t+1}^T, Y_{t+1}^T). \tag{15}$$

As we can see, the auxiliary function $Q(X_1^T, X_{1,p}^T)$ is much simpler than $p_T(X_1^T, Y_1^T)$. In the majority of practical cases it is not difficult to find a maximum of the auxiliary function. For example, if source coded speech signal, which is modeled by an HMM with the mixture of Gaussian state-conditional PDFs, is transmitted over the slowly fading channel with additive Gaussian noise, it is possible to find the closed form solution of equation (11). [14] However, the matrix $\mathbf{P}(X_t, Y_t)$ size might be large. The previous equations dimensionality can be reduced if this matrix has a special form.

5. MAP DECODING OF TCM SIGNALS ON CHANNELS WITH MEMORY

For a TCM system with an i.i.d information sequence, equation (5) takes the form

$$\mathbf{P}(X_t,Y_t) = \left[p_{S_t^{(s)}S_{t+1}^{(s)}}\mathbf{P}_c(Y_t \mid X_t)\right] \tag{16}$$

where

$$p_{S_t^{(s)}S_{t+1}^{(s)}} = \begin{cases} Pr(I_t) & \text{if} \quad S_{t+1}^{(s)} = f_t(S_t^{(s)},I_t) \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

which means that the matrix $\mathbf{P}(X_t,Y_t)$ is sparse. In this case, we can write

$$p_T(I_1^T,Y_1^T) = \boldsymbol{\pi}_c \prod_{t=1}^{T} p_{S_t^{(s)}S_{t+1}^{(s)}}\mathbf{P}_c(Y_t \mid X_t)\mathbf{1}.$$

Here and in the following equations, $\boldsymbol{\pi}_c$ is a vector of the initial probabilities of the channel states, $X_t = g_t(S_t^{(s)},I_t)$, and the product is taken along the state trajectory $S_{t+1}^{(s)} = f_t(S_t^{(s)},I_t)$ for $t=1,2,...,T$. Thus, for the TCM system, instead of a large sparse matrix $\mathbf{P}(X_t,Y_t)$, we can use smaller matrices $\mathbf{P}_c(Y_t \mid X_t)$ for computing the APP.

If we assume that all the information symbols are equiprobable, then the MAP estimate is equivalent to the ML estimate:

$$\hat{I}_1^T = \underset{I_1^T}{argmax}\,\boldsymbol{\pi}_c \prod_{t=1}^{T}\mathbf{P}_c(Y_t \mid X_t)\mathbf{1}.$$

The auxiliary function can be also written in terms of the smaller matrices:

$$Q(I_1^T,I_{1,p}^T) = \sum_{t=1}^{T}\sum_{i=1}^{n_c}\sum_{j=1}^{n_c}\gamma_{t,ij}(I_{1,p}^T)\log p_{c,ij}(Y_t|X_t) + C \tag{18}$$

where $X_t = g_t(S_t^{(s)},I_t)$ and

$$\gamma_{t,ij}(I_{1,p}^T) = \alpha_i(Y_1^{t-1}|I_{1,p}^{t-1})p_{c,ij}(Y_t|X_{t,p})\beta_j(Y_{t+1}^T|I_{t+1,p}^T) \tag{19}$$

$\alpha_i(Y_1^{t-1}|I_{1,p}^{t-1})$ and $\beta_j(Y_{t+1}^T|I_{t+1,p}^T)$ are the elements of the forward and backward probability vectors

$$\alpha(Y_1^t|I_1^t) = \boldsymbol{\pi}_c\prod_{i=1}^{t}\mathbf{P}_c(Y_i|X_i) \quad \text{and} \quad \beta(Y_1^t|I_1^t) = \prod_{i=t}^{T}\mathbf{P}_c(Y_i|X_i)\mathbf{1} \tag{20}$$

which can be computed by the forward and backward algorithms, respectively.

It follows from equation (18) that we can apply the Viterbi algorithm with the branch metric

$$m(I_t) = \sum_{i=1}^{n_c}\sum_{j=1}^{n_c} \gamma_{t,ij}(I_{1,p}^T)\log p_{c,ij}(Y_t|X_t), \quad t=1,2,\ldots,T \tag{21}$$

to find a maximum of $Q(I_1^T, I_{1,p}^T)$ which can be interpreted as a longest path leading from the zero state to one of the states $S_T^{(s)}$. (Note that we are considering only the *encoder* trellis.) It is convenient to use the Viterbi algorithm in the backward direction to combine it with the forward-backward algorithm. In the forward direction, we compute and save in memory $\boldsymbol{\alpha}(Y_1^t|I_{1,p}^t)$ for $t=1,2,\ldots,T$; in the backward direction, we compute recursively $\boldsymbol{\beta}(Y_{t+1}^T|I_{t+1,p}^T)$ for $t=T-1,T-2,\ldots,1$, then we compute $\gamma_{t,ij}(I_{1,p}^T)$ and, for each encoder state at time t, determine the longest path starting from this state with branch metric $m(I_t)$ and the corresponding input sequence generating this path. Thus, the described EM algorithm can be summarized as follows:

1. Select initial sequence $I_{1,0}^T = I_{1,0}, I_{2,0}, \ldots, I_{T,0}$

2. *Forward part:*

   Set $\boldsymbol{\alpha}(Y_1^0|I_1^0) = \boldsymbol{\pi}$ and for $t=1,2,\ldots,T$ compute $X_{t,p} = g_t(S_t^{(s)}, I_{t,p})$,

$$\boldsymbol{\alpha}(Y_1^t|I_{1,p}^t) = \boldsymbol{\alpha}(Y_1^{t-1}|I_{1,p}^{t-1})\mathbf{P}_c(Y_t|X_{t,p}) \tag{22}$$

3. *Backward part:*

   Set $\boldsymbol{\beta}(Y_{T+1}^T|I_{T+1,p}^T) = \mathbf{1}$ and the survivors [4] lengths $L(S_T^{(s)}) = 0$ for $S_T^{(s)} = 1,2,\ldots,n_s$.

   For $t=T,T-1,\ldots,1$ compute $X_t = g_t(S_t^{(s)}, I_t)$,

$$\gamma_{t,ij}(I_{1,p}^T) = \alpha_i(Y_1^{t-1}|I_{1,p}^{t-1})p_{c,ij}(Y_t|X_{t,p})\beta_j(Y_{t+1}^T|I_{t+1,p}^T)$$

$$L(S_t^{(s)}) = \max_{I_t}\{L[f_t(S_t^{(s)}, I_t)] + m(I_t)\}$$

$$\hat{I}_t(S_t^{(s)}) = \underset{I_t}{argmax}\{L[f_t(S_t^{(s)}, I_t)] + m(I_t)\}$$

$$\boldsymbol{\beta}(Y_t^T|I_{t,p}^T) = \mathbf{P}_c(Y_t|X_{t,p})\boldsymbol{\beta}(Y_{t+1}^T|I_{t+1,p}^T) \tag{23}$$

4. Reestimate the information sequence:

$$I_{t,p+1} = \hat{I}_t(\hat{S}_t), \quad \hat{S}_{t+1} = f_t(\hat{S}_t, I_{t,p+1}), \quad t=1,2,\ldots,T$$

   where $\hat{S}_1 = 0$

5. If $I_{t,p+1} \neq I_{t,p}$, go to step 2; otherwise decode the information sequence as $I_{1,p+1}^{T}$.

Note that if the terminal state is forced to be zero, then we set $L(S_T^{(s)}) = -\infty$ for $S_T^{(s)} \neq 0$.

It is clear from the algorithm's description that the number of operations required by the algorithm is $\kappa = N_I \cdot N_{FB} \cdot N_V$ where N_{FB} is the number of operations required by the forward-backward algorithm, [1] N_V is the number of operations required by the Viterbi algorithm, [4] and N_I is the number of iterations. The number of iterations is a random number which depends on the initial approximation of the decoded sequence. Because of the discrete nature of the decoded sequence, the number of iterations is usually small. In our simulation study, for the code described in Sec. 5.2 and channel of Ref. [10], we compared the EM algorithm with the exhaustive search (for short sequences). It took less than three iterations for the algorithm to converge for various initial guesses.

The forward-backward algorithm requires saving in memory all the forward probability vectors. The backward part of the algorithm starts only after the whole sequence Y_1^T has been received. This can cause a significant problem if the sequence is long. The problem can be solved by using an approximate forward-only fixed-lag algorithm. [21] By combining the fixed-lag algorithm with the above EM algorithm it is possible to perform the decoding in the forward-only fashion. The combined algorithm lends itself to parallelization and pipelining.

This algorithm can be applied to a general IOHMM. For the model special cases the branch metric can be simplified. In particular, if the model parameters are given by equation (6), we can write [23]

$$m(I_t) = \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) b_j(Y_t|X_t), \quad t=1,2,\ldots,T. \tag{24}$$

where

$$\gamma_{t,i}(I_{1,p}^T) = \alpha_i(Y_1^t|I_{1,p}^t)\beta_i(Y_{t+1}^T|I_{t+1,p}^T)$$

For the fading channel with AWGN the algorithm can be simplified. Substituting equation (8) into equation (18) we conclude that the maximization of $Q(I_1^T, I_{1,p}^T)$ is equivalent to minimization of the following quadratic form

$$R(I_1^T, I_{1,p}^T) = \sum_{t=1}^{T} \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) \parallel Y_t - a_i X_t \parallel^2 \tag{25}$$

which can be accomplished by the Viterbi algorithm with the branch metric

$$m(I_t) = \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) \parallel Y_t - a_i X_t \parallel^2, \quad t = 1,2,\ldots,T. \tag{26}$$

For a PSK modulated sequence, we can write

$$R(I_1^T, I_{1,p}^T) = -2 \sum_{t=1}^{T} \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) \mathrm{Re}\{X_t Y_t^* a_i\} + C \tag{27}$$

where C is independent of I_1^T. Thus, we can use a simpler metric

$$\mu(I_t) = \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) \mathrm{Re}\{X_t Y_t^* a_i\} \tag{28}$$

in the Viterbi algorithm.

To improve the algorithm's convergence rate, it is important to select a good initial approximation of the decoded sequence. This can be achieved by using one of the suboptimal decoding algorithms. For example, we can use the symbol MAP estimate according to equation (3), the Viterbi algorithm for the most probable path, [15] or an algebraic decoding algorithm.

To illustrate the algorithm applications, we consider two examples.

5.1 Map Decoding of Block Codes

It is well known [1,26] that block codes can be interpreted as trellis codes in the following way. Let $\mathbf{H} = [\ h_1\ h_2\ ,\ldots,\ h_N\ ]$ be a parity check matrix of a linear (N,K) code. Define the encoder states recursively as

$$S_0 = 0, \qquad S_t = S_{t-1} + I_t h_t. \tag{29}$$

Since a codeword I_1^N is in the null-space of $\mathbf{H}$ ($I_1^N \mathbf{H}' = \sum_{t=0}^{N} I_t h_t = 0$), the trellis encoder needs to keep only the trajectories leading to state $S_N = 0$. If the code is systematic, the first K symbols are the information symbols I_1^K given by the source. The parity check symbols I_{K+1}^N are uniquely defined by the path leading from state S_K to state $S_N = 0$. If the source is binary, there are only

two possible transitions from the states corresponding to information bits.

Equation (29) has the form of the first equation of (9). In block-code-based TCM, segments of a codeword X_1^N are mapped into the modulator constellation. If we have a discrete channel, the modulator is considered to be a part of the channel and the encoded symbols are transmitted directly to the channel. In this case, the second equation of (9) takes the form $X_t = I_t$. Thus, both equations (9) are satisfied and we can apply the EM decoding algorithm described in this section if the channel is modeled by an HMM.

To be more specific, consider a (5,3) block code with the parity check matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

The code trellis diagram is depicted in Fig. 1. This code has four states: (00), (01), (10), and (11). The encoder output bits mark the corresponding state transition edges: the (horizontal) transitions between the same states correspond to 0 and transitions between different states correspond to 1. For example, for the first observation Y_1 we have $h_1 = (1,0)'$ which is the first column of the matrix **H**.

Suppose that the channel is described by the Gilbert-Elliott model [3,10] represented by equations (6) and (7). According to these equations, we have

$$\mathbf{P}(0|0) = \mathbf{P}(1|1) = \begin{bmatrix} p_{11}(1-b_1) & p_{12}(1-b_2) \\ p_{21}(1-b_1) & p_{22}(1-b_2) \end{bmatrix}, \quad \mathbf{P}(0|1) = \mathbf{P}(1|0) = \begin{bmatrix} p_{11}b_1 & p_{12}b_2 \\ p_{21}b_1 & p_{22}b_2 \end{bmatrix}.$$

Let us assume that the sequence $Y_1^5 = 01010$ was received. As an initial approximation of the decoded sequence we choose the closest to the received sequence Y_1^5 codeword $X_{1,0}^5 = 01011$. For this codeword, we compute

$$\boldsymbol{\alpha}(Y_1|0) = \boldsymbol{\pi}_c\mathbf{P}_c(0|0), \ \boldsymbol{\alpha}(Y_1^2|01) = \boldsymbol{\alpha}(Y_1|0)\mathbf{P}_c(1|1), \ \boldsymbol{\alpha}(Y_1^3|010) = \boldsymbol{\alpha}(Y_1^2|01)\mathbf{P}_c(0|0)$$

$$\boldsymbol{\alpha}(Y_1^4|0101) = \boldsymbol{\alpha}(Y_1^3|010)\mathbf{P}_c(1|1), \quad \boldsymbol{\alpha}(Y_1^5|01011) = \boldsymbol{\alpha}(Y_1^4|0101)\mathbf{P}_c(0|1).$$

This completes the forward part. In the backward part, we compute

$$L(S_4=00) = m(X_5=0) = \gamma_{5,1}\log(1-b_1)+\gamma_{5,2}\log(1-b_2)$$

$$L(S_4=01) = m(X_5=1) = \gamma_{5,1}\log b_1+\gamma_{5,2}\log b_2$$

where $\gamma_{5,i} = \alpha_i(Y_1^5|01011)\beta_i$ and $\beta_i=1$. Then we compute $\boldsymbol{\beta}(Y_5|1) = \mathbf{P}_c(0|1)\mathbf{1}$ and

$$L(S_3=00) = m(X_4=0)+m(X_5=0), \quad L(S_3=01) = m(X_4=0)+m(X_5=1)$$

$$L(S_3=10) = m(X_4=1)+m(X_5=0), \quad L(S_3=11) = m(X_4=1)+m(X_5=1).$$

In the next step, we need to apply the Viterbi algorithm: Choose $X_3=0$ if

$$L(S_2=00) = L(S_3=00)+m(X_3=0) > L(S_3=01)+m(X_3=1).$$

Otherwise, we choose $X_3=1$ and $L(S_2=00) = L(S_3=01)+m(X_3=1)$, and so on. At the end of this iteration, we find the longest path connecting the nodes at $t=0$ and $t=5$. The corresponding encoder output sequence is the next approximation of the decoder output. The process continues until two consecutive iterations deliver the same result.

5.2 Map Decoding of Convolutional Codes

As a second example, let us consider a convolutional code. The encoder operation can be described by the following equations [5]

$$\begin{aligned} S_{t+1}^{(s)} &= S_t^{(s)}\mathbf{A} + I_t\mathbf{B} \\ \Xi_t &= S_t^{(s)}\mathbf{C} + I_t\mathbf{G} \end{aligned} \qquad t=1,2,... \tag{30}$$

where S_t is the encoder state vector and Ξ_t is the encoder output vector. The encoder initial state is usually selected as $S_1 = 0$.

In the TCM standard implementation, the encoded symbols are transformed by a memoryless mapper to produce a symbol in the modulator constellation:

$$X_t = F(\Xi_t) = F(S_t^{(s)}\mathbf{C} + I_t\mathbf{G}) = f_t(S_t^{(s)},I_t). \tag{31}$$

As we can see, equations (30) and (31) have the form of equations (9). If we assume that symbols X_t are transmitted over a channel represented by equation (6), then we can apply the EM algorithm described in this section.

To be more specific, suppose that the source is binary, memoryless, and symmetric

($p(0) = p(1) = 0.5$). The source bits are coded by the rate 1/2 convolutional encoder shown in Fig. 2.

If we denote the encoder output vector as $\Xi_t = [\xi_{t1} \quad \xi_{t2}]$, then according to this figure we have

$$\begin{aligned} S_{t+1}^{(s)} &= S_t^{(s)}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + I_t[1 \quad 0] \\ \Xi_t &= S_t^{(s)}\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} + I_t[1 \quad 1] \end{aligned} \tag{32}$$

where the encoder state is defined by the contents of its shift registers which is (see Fig. 2) $S_t^{(s)} = [I_{t-1}, I_{t-2}]$ and its state transition probability matrix has the form

$$\mathbf{P}_s = \begin{bmatrix} 0.5 & 0 & 0.5 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0 & 0.5 \end{bmatrix}$$

The encoder output symbols are mapped to the QPSK constellation which consists of four symbols: $X^{(0)} = (-A,-A)$, $X^{(1)} = (-A,A)$, $X^{(2)} = (A,-A)$, and $X^{(3)} = (A,A)$. Equation (16) takes the form

$$\mathbf{P}(I=0,Y)=0.5\begin{bmatrix} \mathbf{P}_c(Y|X^{(0)}) & 0 & 0 & 0 \\ \mathbf{P}_c(Y|X^{(3)}) & 0 & 0 & 0 \\ 0 & \mathbf{P}_c(Y|X^{(2)}) & 0 & 0 \\ 0 & \mathbf{P}_c(Y|X^{(1)}) & 0 & 0 \end{bmatrix} \quad \mathbf{P}(I=1,Y)=0.5\begin{bmatrix} 0 & 0 & \mathbf{P}_c(Y|X^{(3)}) & 0 \\ 0 & 0 & \mathbf{P}_c(Y|X^{(0)}) & 0 \\ 0 & 0 & 0 & \mathbf{P}_c(Y|X^{(1)}) \\ 0 & 0 & 0 & \mathbf{P}_c(Y|X^{(2)}) \end{bmatrix} \tag{33}$$

Similarly to the previous example, we can apply the EM algorithm of this section to decoding of a terminated convolutional code. If a code sequence is too long, an approximate fixed-lag algorithm can be applied in a forward-only fashion. [21] If the channel is described by equation (6), we can apply the EM algorithm whose maximization part is performed using the Viterbi algorithm with the branch metric (24).

However, if a bit-level model is described by equation (6), it does not mean that the encoded symbol model has this form. In this case we need to apply a more complex metric (21). To be more specific, suppose that the channel is described by the Gilbert-Elliott model with parameters [3,10]

$$\mathbf{P}_c = \begin{bmatrix} 0.997 & 0.003 \\ 0.034 & 0.966 \end{bmatrix}, \quad b_1 = 0.001, \quad b_2 = 0.16.$$

Thus, according to equation (6), we have

$$\mathbf{P}_c(0) = \begin{bmatrix} 0.996003 & 0.002520 \\ 0.033966 & 0.811440 \end{bmatrix}, \quad \mathbf{P}_c(1) = \begin{bmatrix} 0.000997 & 0.000480 \\ 0.000034 & 0.154560 \end{bmatrix}.$$

Suppose that the code starts and terminates at the state 00 and has three information bits. The encoder trellis diagram along with its output/input symbols are depicted in Fig. 3. For every information bit, the encoder produces a two-bit symbol which is presented in decimal form in Fig. 3. Thus, we have the following matrix probabilities of errors in the received two-bit symbols:

$$\mathbf{P}(0) = \mathbf{P}_c^2(0) = \begin{bmatrix} 0.992108 & 0.004555 \\ 0.061392 & 0.658520 \end{bmatrix}, \quad \mathbf{P}(1) = \mathbf{P}_c(0)\mathbf{P}_c(1) = \begin{bmatrix} 0.000993 & 0.000868 \\ 0.000061 & 0.125432 \end{bmatrix},$$
$$\mathbf{P}(2) = \mathbf{P}_c(1)\mathbf{P}_c(0) = \begin{bmatrix} 0.001009 & 0.000392 \\ 0.005284 & 0.125416 \end{bmatrix}, \quad \mathbf{P}(3) = \mathbf{P}_c^2(1) = \begin{bmatrix} 0.000001 & 0.000075 \\ 0.000005 & 0.023889 \end{bmatrix}.$$

These matrices do not satisfy equation (6), therefore, we cannot use the simplified metrics as in the previous examples and must use a more complex metric (21). Equation (16) takes the form

$$\mathbf{P}(I=0,Y)=0.5\begin{bmatrix} \mathbf{P}_c(Y) & 0 & 0 & 0 \\ \mathbf{P}_c(Y\oplus 3) & 0 & 0 & 0 \\ 0 & \mathbf{P}_c(Y\oplus 2) & 0 & 0 \\ 0 & \mathbf{P}_c(Y\oplus 1) & 0 & 0 \end{bmatrix} \quad \mathbf{P}(I=1,Y)=0.5\begin{bmatrix} 0 & 0 & \mathbf{P}_c(Y\oplus 3) & 0 \\ 0 & 0 & \mathbf{P}_c(Y) & 0 \\ 0 & 0 & 0 & \mathbf{P}_c(Y\oplus 1) \\ 0 & 0 & 0 & \mathbf{P}_c(Y\oplus 2) \end{bmatrix} \tag{46}$$

where $\oplus$ denotes bitwise exclusive-or operation. Suppose that we received $Y_1^5 = (0,3,2,1,0)$. As an initial guess for the transmitted codeword we choose $X_0 = 0,0,0,0,0$. Using the algorithm described on page 9, we can find a maximum of $Q(I_1^T, I_{1,p}^T)$. Obviously, it is equivalent to finding a minimum of $-Q(I_1^T, I_{1,p}^T)$ which can be achieved using the Viterbi algorithm by finding the shortest path on the trellis using negative metric in equation (21). We obtained after first iteration the following lengths of the shortest paths connecting each node of the trellis with its terminal node:

state	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
0	$0.074\cdot 10^{-3}$	$0.0618\cdot 10^{-3}$	$0.0783\cdot 10^{-3}$	$0.0443\cdot 10^{-3}$	$0.0101\cdot 10^{-3}$
1			$0.0784\cdot 10^{-3}$	$0.0442\cdot 10^{-3}$	$0.0731\cdot 10^{-3}$
2		$0.0896\cdot 10^{-3}$	$0.0511\cdot 10^{-3}$	$0.1344\cdot 10^{-3}$	
3			$0.0511\cdot 10^{-3}$	$0.0800\cdot 10^{-3}$	

The decoded sequence $\hat{I}_1^5 = 0,1,0,0,0$ and the corresponding transmitted sequence $\hat{X}_1^5 = 0,3,2,3,0$ After second iteration we obtained the following length

state	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
0	0.0248	0.0229	0.0344	0.0177	0.0020
1			0.0341	0.0171	0.0289
2		0.0382	0.0204	0.0567	
3			0.0204	0.0319	

and the same decoded sequence as in the previous iteration $\hat{I}_1^5 = 0,1,0,0,0$.

As a test, we performed an exhaustive search which gave the same result with a maximum likelihood value of 0.940201. For the HMM fading channel with the AWGN we can use metric (28).

6. CONCLUSION

We have demonstrated that if an information source, encoder, and communication channel are modeled by IOHMMs, then MAP decoding can be realized using the EM algorithm. The expectation part of the algorithm is performed using the forward-backward algorithm while the maximization part is accomplished using the Viterbi algorithm. The EM algorithm is robust and, in contrast with some other iterative algorithms, it converges to the APP maximum in all practical cases. [2] Because the set of transmitted symbols is discrete, the number of necessary iterations is usually small which was confirmed by a direct simulation.

7. ACKNOWLEDGMENT

Helpful comments and suggestions of the associate editor Dr. Steven S. Pietrobon and the anonymous reviewers are gratefully acknowledged.

References

[1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory,* vol IT-20, pp. 284-287, Mar. 1974.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via

the EM algorithm," *J. R. Statist. Soc.*,vol. 76, 1977, pp 341-353.

[3] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.,* vol. 42, Sept. 1963, pp. 1977-1997.

[4] G.D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE,* vol. 61, pp. 268-278, Mar. 1973.

[5] G.D. Forney, Jr., "Convolutional codes I: algebraic structure," *IEEE Trans. Inform. Theory,* vol IT-16, Nov. 1970, pp. 720-738.

[6] G.D. Forney, Jr., R.G. Gallager, G.R. Lang, F.M. Longstaff, and S.U. Qureshi, "Efficient modulation for band-limited channels," *IEEE J. Select. Areas Commun.,* vol. SAC-2, pp. 632-647, Sept. 1984.

[7] J. Garcia-Frias and J. D. Villasenor, "Exploiting binary Markov channels with unknown parameters in turbo coding," *GLOBECOM'98 Conf. Record* Sydney, 1998, pp. 3244-3249.

[8] C. N. Georghiades and J. C. Han, "Sequence estimation in the presence of random parameters via the EM algorithm," *IEEE Trans. Comm.*, vol 45, no. 3, March, 1997.

[9] C. N. Georghiades, "Maximum-likelihood detection in the presence of interference,"*Proc. IEEE Symp. Inf. Theory*, Aug. 1998, p. 344.

[10] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, Sept. 1960, pp. 1253—1266.

[11] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Trans. Inform. Theory,* col IT-23, pp. 371-377, May 1977.

[12] L. N. Kanal and A. R. K. Sastry, "Models for channels with memory and their applications to error control," *Proc. of the IEEE,* vol. 66, no. 7, July 1978.

[13] J. H. Kang, W. E. Stark, and A. O. Hero, "Turbo codes for fading and burst channels," *GLOBECOM'98 Communications Miniconference,* Sydney, 1998, pp. 183-188.

[14] P. Ligdas, W. Turin, and N. Seshadri, "Statistical methods for speech transmission using hidden Markov models," in *Proc. 31th Conf. on Inf. Sci. and Sys.*, pp. 546-551, March 1997.

[15] N. Merhav and Y. Ephraim, "Hidden Markov modeling using a dominant state sequence with applications to speech recognition," *Computer, Speech and Language*, vol. 5, 1991, pp. 327-339.

[16] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[17] S.O. Rice, "Distribution of the duration of fades in radio transmission: Gaussian noise model," *Bell Syst. Tech Journ.*, vol. 37, May 1958, pp. 581-635.

{18] M. Sajadieh, F.R. Kschischang, and A. Leon-Garcia, "A block memory mode l for correlated Rayleigh fading channels," *Proc. IEEE Int. Conf. Commun.*, June 1996, pp. 282-286.

[19] S. Sivaprakasam and K. S. Shanmugan, "A forward-only recursion based HMM for modeling burst errors in digital channels," GLOBECOM, 1995.

[20] F. Swarts and H.C. Ferreira, "Markov characterization of channels with soft decision outputs," *IEEE Trans. Commun.*, vol. 41, May 1993, pp. 678-682.

[21] W. Turin, "MAP symbol decoding in channels with error bursts," in *Proc. 33 Conf. on Inf. Sci. and Sys.*, pp. 674-679, March 1999.

[22] W. Turin and R. van Nobelen, "Hidden Markov modeling of flat fading channels," *IEEE JSAC*, vol. 16, no. 9, Dec. 1998, pp.1809-1817.

[23] W. Turin, *Digital Transmission Systems: Performance Analysis and Modeling*, McGraw-Hill, New York, 1998.

[24] G. Ungerboeck, "Channel coding with multilevel phase signals," *IEEE Trans. Inform. Theory*, col IT-28, pp.55-67, Jan. 1982.

[25] H.S. Wang and N. Moayeri, "Finite-state Markov channel - a useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, Feb. 1995, pp.163-171.

[26] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using trellis," *IEEE Trans. Inform. Theory*, vol IT-24, pp. 76-80, Jan. 1978.

[27] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, John Wiley & Sons, New York, pp. 213-214, 1965.

[28] L. M. Zeger and H. Kobayashi, "Sequence estimation in GSM using the EM algorithm," *Conf. on Inform. Sci. and Syst.*, March, 1998.